StarTeam User's Guide

StarTeam®

Borland®
Excellence Endures™

Borland Software Corporation
100 Enterprise Way
Scotts Valley, California 95066-3249
www.borland.com

Borland Software Corporation may have patents and/or pending patent applications covering subject matter in this document. Please refer to the product CD or the About dialog box for the list of applicable patents. The furnishing of this document does not give you any license to these patents.

COPYRIGHT © 1995–2005 Borland Software Corporation. All rights reserved. All Borland brand and product names are trademarks or registered trademarks of Borland Software Corporation in the United States and other countries. All other marks are the property of their respective owners.

For third-party conditions and disclaimers, see the Release Notes on your product CD.

Printed in the U.S.A. Updated: June 21, 2005

Contents

Chapter 1
Preface 1
Contacting Borland Support . . . 1
Documentation Conventions . . . 2

Chapter 2
Introducing the Application 3
What's New? . . . 3
Application Variations . . . 4
Using This Manual . . . 4
Understanding the Application . . . 5
Using the Project View Window . . . 7
Using the Folder Hierarchy . . . 7
Using the Upper Pane . . . 8
Using the Lower Pane . . . 10
Selecting the Detail Tab . . . 10
Selecting the History Tab . . . 10
Selecting the Label Tab . . . 10
Selecting the Link Tab . . . 11
Selecting the Reference Tab . . . 11
Using the Title Bars . . . 12
Using the Status Bar . . . 12
Using Version Control . . . 13
Using Defect Tracking . . . 13
Using Requirements . . . 15
Using Tasks . . . 15
Using Topics . . . 16

Chapter 3
Using Projects and Views 19
Adding Access to a Server . . . 19
Opening an Existing Project . . . 20
Transferring Data between Your Workstation and the Server . . . 21
Using View Shortcuts . . . 22
Switching Views . . . 22
Configuring a View to a Point in the Past . . . 22
Controlling File Status Information . . . 24
Using Alternate Working Folders . . . 24
Using View Profiles . . . 25
Controlling Folders in Profiles . . . 25
Controlling EOL Characters and Path Case Sensitivity in Profiles . . . 26
Reviewing View Properties . . . 26
Refreshing Views . . . 27
Reviewing Connection Properties . . . 27
Logging Off the Application . . . 27

Chapter 4
Using the Toolbar 29
Starting the Toolbar . . . 30
Opening the Toolbar Window . . . 30
Specifying the Toolbar User Interface . . . 31
Displaying the Toolbar Window at Startup . . . 31
Hiding the Taskbar Button While the Toolbar Window is Minimized . . . 32
Hiding the Toolbar System Tray Icon . . . 32
Hiding the Toolbar User Interface . . . 33
Logging Onto Servers . . . 33
Viewing your Cached Server/User Associations . . . 33
Logging On As a Different User . . . 34
Using the Same User Name and Password . . . 34
Launching a Tool . . . 35
Customizing the Toolbar . . . 35
Adding a Tool . . . 35
Modifying the Properties of a Tool . . . 36
Deleting a Tool . . . 37
Refreshing the Toolbar . . . 37
Managing the Toolbar Window . . . 37
Collapsing and Expanding the Toolbar Window . . . 38
Resizing the Toolbar Window . . . 38
Minimizing the Toolbar Window . . . 38
Closing the Toolbar Window . . . 38
Displaying the Toolbar Help Information . . . 38
Displaying the Toolbar About Dialog . . . 39
Exiting the Toolbar . . . 39

Chapter 5
Managing Folders 41
Understanding the Folder Hierarchy . . . 41
Adding New Folders . . . 42
Adding Existing Folders . . . 43
Locating Working Folders . . . 43
Using an Alternate Working Folder for the View . . . 44
Adding Folders to a View . . . 44
Changing a Folder Name or Description . . . 46
Excluding Files from Folders . . . 46
Creating Folders for Other Operating System Files . . . 46
Using an Alternate Working Folder . . . 47
Changing a Working Folder . . . 48
Using Folder Properties in a View Profile . . . 49

Chapter 6
Managing Data in the Upper Pane 53
Filter Performance . . . 53
Controlling the Columns . . . 53
Sorting and Grouping the Data . . . 54
Using Queries . . . 55
Applying an Existing Query . . . 56
Creating a Query . . . 56
Creating a Simple Query . . . 57
Creating a More Complex Query . . . 58
Troubleshooting a Query . . . 59
Editing a Query . . . 60
Copying a Query . . . 60
Deleting a Query . . . 60
Using Filters . . . 61
Applying an Existing Filter . . . 62
Creating a Filter . . . 62
Editing a Filter . . . 63
Copying a Filter . . . 64
Resetting a Filter . . . 64

Deleting a Filter . . . 65
Setting Access Rights for Queries and Filters . . . 65
Understanding the Fields . . . 65
Relational Operators . . . 65
File Fields . . . 67
Change Request Fields . . . 76
Requirement Fields . . . 84
Task Fields . . . 90
Topic Fields . . . 98
Audit Log Fields . . . 103

Chapter 7
Using Visual Merge and Visual Diff 111
Comparing the Visual Merge and Visual Diff Utilities 111
Using Visual Merge . . . 113
Using the Stand-alone Utility . . . 113
Using Visual Merge . . . 114
Using the Main Window to Merge Text Files . . . 115
Changing File Encodings . . . 115
Searching Text Files . . . 116
Exiting Visual Merge . . . 117
Customizing Visual Merge . . . 118
Using Visual Merge for Binary Files . . . 118
Using Visual Diff . . . 118
Using the Visual Diff Main Window . . . 119
Using the Menu Commands . . . 119
Changing File Encodings . . . 120
Manipulating the Window Splitter Bars with the Mouse . . . 121
Comparing Files and Viewing Differences . . . 121
Searching Text Files . . . 122
Merging or Printing Compared Files . . . 123
Customizing Visual Diff . . . 124

Chapter 8
Managing Files 127
Using a File Under Version Control . . . 127
Using the File List in the Upper Pane . . . 128
Using File Filters . . . 129
Determining the Status of Your Files . . . 130
Adding Files to a View . . . 130
Reviewing or Setting the Executable Bit . . . 132
Opening Files from the Application . . . 133
Editing Files from the Application . . . 133
Updating File Status . . . 134
Checking In Files . . . 134
Checking Out a Revision . . . 136
Understanding the Effects of Status on Check-ins and Check-outs . . . 139
Reviewing the Content of a Revision . . . 140
Comparing File Contents . . . 140
Merging Files . . . 141
In the Windows Client . . . 141
In the Cross-Platform Client . . . 142
Using Keywords in Files . . . 144
Modifying File Properties . . . 145
Setting File Options . . . 147

Chapter 9
Using Requirements 149
Displaying Existing Requirements in the Upper Pane . . . 149
Benefits of Using the Tree Format . . . 151
Benefits of Using the List Format . . . 151
Using Requirement Filters . . . 151
Using the Lower Pane . . . 151
Creating a Requirement or Child Requirement . . . 152
Exporting Requirements from CaliberRM . . . 153
Importing Requirements from CaliberRM . . . 153
Displaying a CaliberRM Requirement in StarTeam . 154

Chapter 10
Tracking Change Requests 155
Change Request Tracking System Model . . . 155
Built-in Workflow . . . 157
Displaying Existing Change Requests in the Upper Pane . . . 158
Using Change Request Filters . . . 160
Using Existing Change Requests . . . 160
Using the Lower Pane . . . 161
Submitting a Change Request . . . 161
Assigning Change Requests . . . 163
Resolving Open Change Requests . . . 163
Verifying Resolved Change Requests . . . 164
Closing Verified Change Requests . . . 165
Summary . . . 166
Using Reports and Charts . . . 168

Chapter 11
Managing Tasks 171
Using Tasks in the Upper Pane . . . 171
Benefits of Using the Tree Format . . . 172
Benefits of Using the List Format . . . 173
Using Task Filters . . . 173
Creating a Task or Subtask . . . 174
Assigning Task Resources . . . 176
Estimating the Time Required to Complete a Task . 176
Adding Work to a Task or Subtask . . . 177
Adding Notes to a Task . . . 178
Setting Custom Property Fields for Tasks . . . 178
Including Attachments . . . 178
Including Comments for Task Revisions . . . 179
Modifying Task Properties . . . 179

Chapter 12
Managing Topics 181
Using Topics in the Upper Pane . . . 182
Benefits of Using the Tree Format . . . 182
Benefits of Using the List Format . . . 183
Using Topic Filters . . . 183
Creating a Topic . . . 184
Replying to a Topic or Response . . . 185
Modifying Topic and Response Properties . . . 185

Chapter 13
Using the Audit Log **187**
Using Audit Log Filters . . . 188
Audit Log Events . . . 189

Chapter 14
Performing Generic Operations **191**
Moving Folders and Items Between Project Views . 193
Creating Shortcuts to Items . . . 194
Opening and Copying StarTeam URLs . . . 195
Sharing Folders and Items with Another View . . . 195
Understanding Branching . . . 197
Branching Is Disabled . . . 197
Branching Is Set to Branch On Change . . . 198
Branching Is Not Set to Branch On Change . . . 198
Branching a Folder, File, or Change Request . . . 198
Reconfiguring Folders and Items . . . 199
Locking and Unlocking Item . . . 202
Comparing Properties . . . 203
Using the History List . . . 203
Reviewing a Revision's Properties . . . 204
Editing a Revision's Comment . . . 204
Merging Revisions . . . 205
Finding Items by Field Content . . . 205
Printing Items . . . 206
Sending Items via E-mail . . . 206
Controlling System Tray Notification . . . 208
Marking Items Read or Unread . . . 208
Managing Read/Unread Items . . . 208
Flagging Items . . . 208
Deleting a Folder or Item . . . 209

Chapter 15
Using Process Rules **211**
Using Process Items . . . 211
Understanding Process Item Restrictions . . . 212
Pre-selecting an Active Process Item . . . 212
Linking Files to Process Items . . . 213
Using the Links to a Process Item . . . 213

Chapter 16
Using Labels **215**
Creating View Labels . . . 216
Copying View Labels . . . 217
Creating Revision Labels . . . 219
Copying Revision Labels . . . 219
Freezing Labels . . . 220
Deleting Labels . . . 221
Reviewing Label Properties . . . 221
Labeling a Folder . . . 222
Labeling an Item . . . 223
Access Rights that Apply to Labels . . . 226

Chapter 17
Using Links **227**
Linking Folders and Items . . . 227
Understanding the Columns in the Link List . . . 228
Selecting Specific Revisions for Links . . . 229
Reviewing Link Properties . . . 231
Selecting a Linked Item . . . 232
Working with Files Linked to a Specific Item . . . 232
Working with Files on the Links Pane . . . 233
Deleting Links . . . 233

Chapter 18
Using References **235**
Viewing References . . . 236
Understanding the Initial Reference . . . 238
Understanding References Created by Branching Views 239
Understanding References Created by Adding Items to Views . . . 241
Floating Downwards . . . 242
Floating Upwards . . . 242
Floating Upwards and Downwards . . . 243
Understanding References Created by Manually Sharing Objects . . . 244
Floating Upwards and Downwards . . . 244
Understanding References Created by Moving Objects . 245
Floating Upwards and Downwards . . . 246

Chapter 19
Reporting and Exporting Data **249**
Sample Scenarios for Generating Reports . . . 249
Creating Reports . . . 250
Exporting Data . . . 251
Available Encodings . . . 252
Available Reports . . . 253
Using Report Templates . . . 255
Changing the Fields Specified in the Templates . 257

Chapter 20
Using Charts **259**
Available Charts . . . 259
Creating a Simple Chart . . . 262
Creating a Distribution Chart . . . 262
Creating a Correlation Chart . . . 263
Creating a Time-Series Chart . . . 263
Using the Chart Window in the Windows Client . . . 264
Editing Charts . . . 265
Changing a Chart Type . . . 265
Changing Chart Data . . . 265
Changing Chart Color or Pattern . . . 265
Changing Chart Text . . . 266
Changing Viewing Options for 3D Charts . . . 267
Exploding a Pie Chart . . . 267
Using the Chart Window in the Cross-Platform Client . . 268
Using the Chart Window Toolbar . . . 268
Exporting Chart Images . . . 268
Changing Chart Type . . . 269
Editing Chart Colors . . . 270
Viewing a Chart in 3D or 2D . . . 271
Zooming and Rotating Charts . . . 271
Showing Legends on Charts . . . 272
Showing Horizontal or Vertical Grids on Charts . 272

Editing Titles on Charts 272
Viewing Chart Data 272

Chapter 21
Setting Personal Options 273
Customizing the Application 273
Setting File Options. 275
Managing Read/Unread Items 280
Controlling System Tray Notification 281
Automatic Locking of Items Other than Files 282
Using StarTeamMPX 282
Changing Personal Information. 283

Chapter 22
Using the stcmd Command-line Interface 285
Syntax 286
Common Options 286
Exit Codes 291
Adding Files: stcmd add 292
Adding Folders: stcmd add-folder 293
Adding Projects: stcmd add-project 295
Adding Views: stcmd add-view 297
Applying Labels: stcmd apply-label. 298
Checking in Files: stcmd ci 299
Checking Out Files: stcmd co 301
Deleting Local Files: stcmd delete-local 304
Comparing File Revisions: stcmd diff. 305
Changing Descriptions: stcmd dsc 307
Displaying File History: stcmd hist 308
Creating Labels: stcmd label 309
Locking and Unlocking Files: stcmd lck. 310
Listing Files: stcmd list 311
Creating Working Folder: stcmd local-mkdir 312
Merging Files 312
Moving Status: stcmd move-status 312
Removing Files: stcmd remove. 314
Locking/Unlocking a Server: stcmd server-mode . 314
Setting Personal Options: stcmd set-personal-options 315
Updating File Status: stcmd update-status 316

Chapter 23
Using the bco Command Line Interface 319
Syntax 319

Appendix A
Glossary 327

Index 337

Chapter

1

Preface

This manual provides detailed information on using the application to track and manage changes to files, share files among team members, access prior versions of a file, and other functions. Also, the *StarTeam User Guide* explains how to use change requests, requirements, topics, and tasks. However, depending on your license, not all functionality discussed in this manual will be available to you.

Contacting Borland Support

Borland Software Corporation is committed to providing world-class services in the area of consulting and technical support. We have over 15 years of experience in supporting developers and enterprise customers. Our qualified technical support engineers are prepared to handle your support needs on a case-by-case basis or in an ongoing partnership. Borland provides support worldwide, delivering timely, reliable service to ensure every customer's business success.

For more information about Borland's support services, please see our web site at http://support.borland.com.

From the Web site, you can also access many newsgroups where users exchange information, tips, and techniques. See http://info.borland.com/newsgroups/ for the latest list of free product newsgroups. Also available on the Internet is the Borland Developer Network site at http://community.borland.com. This Borland Community provides access to product specific information, articles, code examples, and news.

When contacting support, be prepared to provide complete information about your environment, the version of the product you are using, and a detailed description of the problem.

For support on third-party tools or documentation, contact the vendor of the tool.

Documentation Conventions

The documentation uses the following conventions.

Select File > Exit	Indicates a menu selection followed by a submenu selection. The greater-than symbol (**>**) separates the commands to be selected from subsequent menus. For example, “Select the File > Exit command” means to select File from the menu bar and then select Exit from the drop-down menu.
Fixed-Space Font	Text appearing in Courier font represents information that you need to type and messages from the system.
italics	Syntax appearing in italics represents information that you replace with the names of your files, child folders, etc. Italics are also used for the names of dialogs and books and for emphasis.
Bold	Syntax appearing in bold represents information that you must use exactly as shown (if you use it).
[ ]	Square brackets surround optional syntax.
|	A vertical bar separates mutually exclusive choices in syntax.
	When this icon appears in the margin, it indicates that the section or procedure applies to the Windows client but not to the Cross-Platform client. (If a section or procedure applies to both clients, no icon appears in the margin.)
	When this icon appears in the margin, it indicates that the section or procedure applies to the Cross-Platform client but not to the Windows client. (If a section or procedure applies to both clients, no icon appears in the margin.)

Note Identifies supplemental information.

Tip Identifies information on alternative procedures or other helpful but nonessential information.

Important Identifies information that is essential to the completion of a task.

Caution Identifies actions that may result in loss of data or procedures that must be followed to ensure that data is *not* lost.

Chapter

2

Introducing the Application

The StarTeam application facilitates both local and remote team collaboration through the use of tightly integrated components commonly used during product development. It enables you to:

- Store everything related to a project in the same place. For example, you can store text and binary files, change requests, and discussions about a project.
- Access information and collaborate via a LAN, WAN, or the Internet.
- Improve project teamwork.
- Make testing easier and save maintenance time because of the tight integration between change requests and code and content files.
- Makes tracking requirements easier because of the tight integration between requirements and code and content files.
- Makes project management easier because the tight integration between tasks and code and content files.
- Keep track of who did what and when.
- Take advantage of true client/server access to data.

The application has Windows and Cross-Platform clients. The Server runs on both Windows and Unix platforms.

What's New?

Check the readme files on your installation CD for the latest information about the application. If you were using the previous release, you can read the *StarTeam New Features Guide*, which describes the new features in this release of the product line.

Application Variations

To use the application, you need both the Server and a client to access that server. Most users have more than one client. The most common clients are explained in the following table.

Table 2.1 Server Clients

Client	Description
Windows client	Original client created for Windows platform; it has complete feature set.
	This client comes with the COM version of the stcmd command line and the Java version of the stcmd command line.
Cross-Platform client	Created in Java for Solaris, Linux, and Macintosh System X platforms; can work on any platform that supports Java; it has a subset of the features available in the Windows client.
	This client comes with the Java version of the stcmd command line
StarDisk	Another client for the Windows platform. This client can be used from Windows Explorer and supports only the file component.
Web Edition	A browser-based client that can be used from any platform; it has a subset of the features available in the Windows client and a few additional features used only by Web Edition.
Integrations	The application has a variety of integrations with Integrated Development Environments (IDEs) and other products, such as Microsoft Project.

This book covers the Windows and Cross-Platform clients. The other clients have their own documentation. For example, the *StarDisk User's Guide* explains how to use StarDisk.

The application's servers and clients are licensed for specific feature sets. Some of the features explained in this guide may not be available to you because of your client's or the server's license.

Using This Manual

This book covers the features most often used by team members. The features most often used by administrators are explained in the *StarTeam Administrator's Guide*. Depending on the role you play as a user, you may need some or all of the information in the *StarTeam Administrator's Guide* as well. For example, although you may not be an administrator, you may still be required to create new projects, new views, and/or set access rights. Details about these operations are in the *StarTeam Administrator's Guide*.

This book provides an introduction to the Windows and Cross-Platform clients. It then covers projects, view, and folders, although not in the same depth as in the *StarTeam Administrator's Guide*.

Next, it covers each of the types of items stored in the application repository: files, change requests, requirements, tasks, topics, and audit events. Depending upon your application license, your company may not have access to all these item types.

Each item-related chapter explains what is unique to the item being covered and explains how to use the standard property dialog for that item.

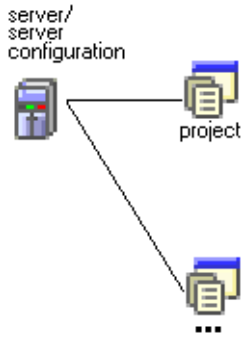
If you use alternate property editors (APEs) instead of the standard property dialogs, some of this information will not apply. Your company may provide you with additional information that explains the use of your APEs. Because APEs are customized forms, they are different at every location.

After the item-related chapters, the manual includes a chapters on performing common operations that relate to more than one item type. If the operation takes only a few paragraphs to explain, it is covered in this chapter. If the operation requires several pages to explain, it is covered in its own chapter. For example, labels and references have chapters of their own.

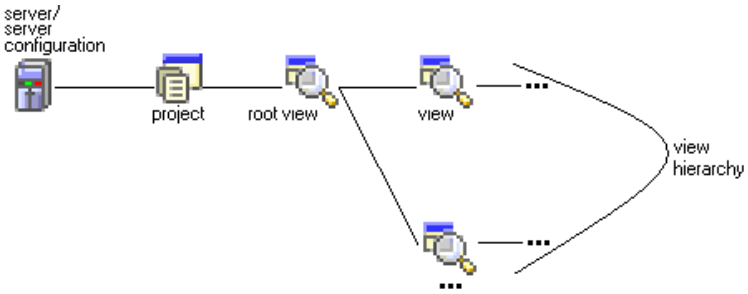
Understanding the Application

The following figures provide an overview of the project structure controlled by an instance of the Server running a specific server configuration.

A server/server configuration can manage any number of projects. A project is a way to group your files according to their purpose. For example, a project may be include all the files related to a software application created by your company or all the files related to a component of that software application.



Each project has one root view and any number of child views.

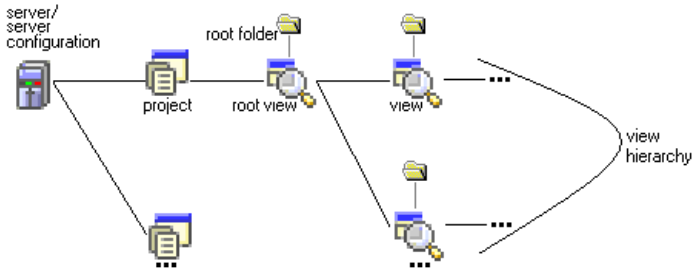


Your administrator will use views to:

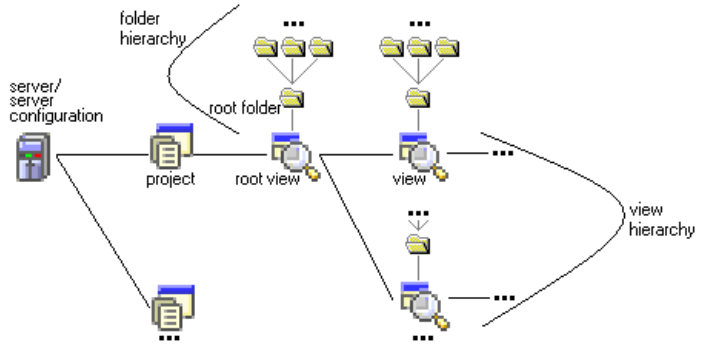
- Provide access to only a subset of the files in a project. For example, a developer may not need to be aware of files stored in that project for marketing purposes and a market researcher might not need to be aware of the source code files. They can both work on the same project, but in different views, each of which is a subset of a the root view. Such views are called reference views.
- Provide a branch for the project. This is done by creating a new view, a branching view, for the project. For example, one view of a project can store the source code for Release 1.0 of a software application. Another view can be for Release 2.0.

  The application also enables you to merge files from these two views. For example, suppose your company needs to create a Service Pack or patch for Release 1.0. If the required code changes have already been added to a source code file in the 2.0 view, you can merge the contents of that file with the contents of the corresponding file in the 1.0 view. Then both files have the required changes.

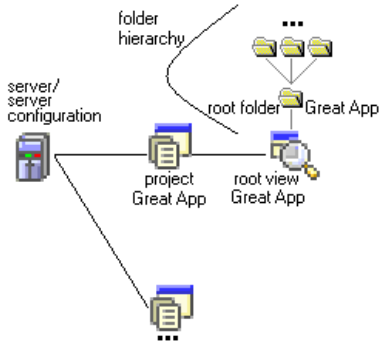
One folder serves as the root folder for the root view and every child view.



A root folder can have any hierarchy of child folders. This hierarchy is called the folder hierarchy.



When you create a project, its root view, and the root view's root folder are created automatically and given the same name as the project. For example, if the project's name is Great App, the root view's name is initially Great App, and the root folder's name is initially Great App. You can change any of these names using the application (if you have the access rights to do so).



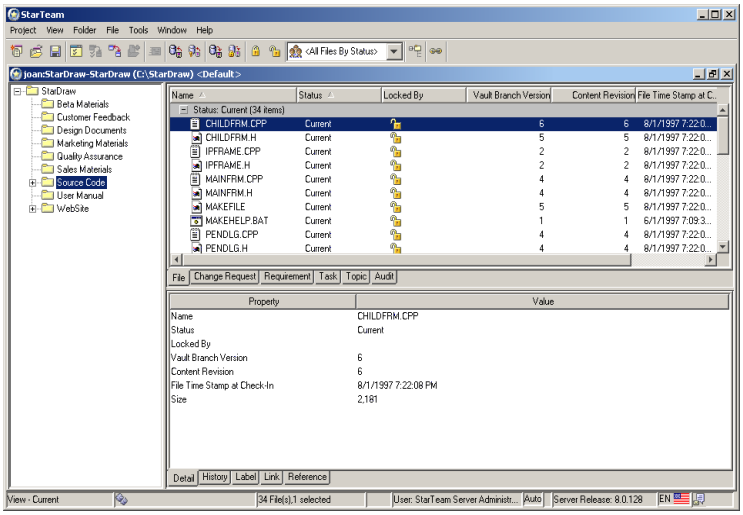
Each application folder has a corresponding working folder. The working folder is the location to which files are checked out your workstation. The path to the working folder may have little or no correlation to the path to the application folder.

For example, assume that the root folder for a view is Great App and it has a child folder named Source Code. Great App might have the working folder C:\Great App and Source Code might have the working folder C:\Great App\Source Code or even E:\VBstuff\sc. The application folder Source Code may be a child folder of the folder

named Great App, but it does not necessarily have a working folder that is a child folder of Great App's working folder.

Using the Project View Window

The project view window displays information about only one project view at a time. However, you can open more than one project view window at a time.



Each window has three panes:

- The left pane displays the folder hierarchy for the current view of the current project.
- The upper pane (on the right) displays data associated with the selected folder. The type of data depends on the tab selected from the upper pane. The application has a tab for each component: the File, Change Request, Task, Topic, and Audit components.
- The lower pane (also on the right) displays more information about the item selected from the upper pane. The type of information depends on the tab selected from the lower pane.

Note Requirements, tasks, and topics can be displayed in the upper pane in either a tree or list format. Files, change requests, and audit entries are always displayed in lists.

Using the Folder Hierarchy

Your administrator is usually in charge of creating projects, different views of those projects, and the folder hierarchy. However, you may occasionally add folders to the hierarchy yourself as well as check files in and out. For these reasons, you need to understand the folder hierarchy and its underlying structure of working folders. For more information, see "Managing Folders" on page 41.

Usually you work with only a few folders at a time in a view created by your system administrator that shows these folders.

Selecting one of these folders is like selecting a folder in Windows Explorer. However, while the Windows Explorer folder hierarchy represents the exact path to a folder on your workstation (or elsewhere), the folder hierarchy is not required to do so. Each application folder is associated with a working folder that does have an exact path on your workstation. Files are worked on, checked out to, and checked in from the working folder. The root folder of the application folder hierarchy may be named ProjectX and have a working folder at C:\companyName\ProjectX.

A folder icon precedes each folder name. Clicking the folder name or the icon displays information associated with that folder in the upper pane. The information displayed depends on the folder selected from the folder hierarchy and the tab selected from the upper pane.

You can expand or collapse branches of the folder hierarchy:

- A plus sign identifies a collapsed branch. Clicking it expands the branch.
- A minus sign identifies an expanded branch. Clicking it collapses the branch.

Double-click project folder names in the folder hierarchy to expand or collapse branches.

After selecting a folder from the folder hierarchy, use the Folder menu to perform actions on that folder or right-click to display a context menu.

You can drag a child folder to move it from one parent folder to another. You can also drag items from the upper pane to a folder in the folder hierarchy (left pane) or to a folder in another folder hierarchy (the left pane of another view window). This moves them from one folder to another or from one view or project to another.

Note You can move an item from one view or project to another only if both views belong to the same project or to projects that store data in the same repository (that is, they use the same server configuration).

To share items in one folder or view with another, press *Ctrl* before dragging.

Using the Upper Pane

Each folder in the left pane forms a one-to-many relationship with the contents of the upper pane, meaning each folder in the left pane can contain data items, such as files and change requests. The upper pane displays this data. The tabs beneath the upper pane determine the type of data displayed. For example, it might display the files associated with the folder selected from the folder hierarchy.

You can manipulate these items using these tabs. Some examples include:

- The File component provides version control for files stored in the application.

  Usually all the files related to a project or product are stored together in the application, with the exception of files that can be compiled from those already in the application. For example, a company may store their source code for a product but any executables. Some companies may store both of these. Still others may store only the executable. For more information, see “Using Version Control” on page 13 and “Managing Files” on page 127.

- The Change Request component provides a defect tracking system.

  Companies need to know what parts of a product are malfunctioning and track how each problem is resolved. For more information, see “Using Defect Tracking” on page 13 and “Tracking Change Requests” on page 155.

- The Task component enables team members to track who has been assigned what tasks and how those tasks are progressing toward completion. For more information, see “Using Requirements” on page 15 and “Managing Tasks” on page 171.

- The Topic component enables team members to store threaded conversations about the project or product. For more information, see “Using Topics” on page 16 and “Managing Topics” on page 181.

- The Audit component logs entries that indicate what operations have been performed, when, and by whom. For more information, see “Using the Audit Log” on page 187.

The information displayed in the upper pane depends on all of the following:

- The folder selected from the folder hierarchy.
- The selected component tab: File, Change Request, Topic, Task, or Audit.

  When you click a tab, a menu corresponding to the selected tab becomes the fourth menu on the menu bar. For example, if you select the Change Request tab, the Change Request menu appears on the menu bar.
- The filter selected from the Filter drop-down list above the upper right pane. Filters are listed in alphanumeric order and their names are *not* case sensitive.
- The All Descendants button, located above the upper right pane, indicates the depth for which the application displays information in the upper right pane. You can also turn All Descendants on and off by selecting it from the File, Change Request, Task, Topic, or Audit menus. You can minimize the main window during an All Descendants operation without stopping that operation.

  When this button is not selected, the application displays information for the selected folder only.

  When this button is selected, the application displays information for the selected folder, its children, its children's children, and so on.

Although the type of data in the upper pane varies based on the tab selected, it is manipulated in similar ways.

The application can display a row for every item of that type in the repository or you can create a query and display rows for only the items whose values in one or more columns match the specified criteria.

You can display a maximum of 60 columns. The columns represent the properties (also called property fields) for each item. For example, file properties include the file's name, status, and size. The columns can be rearranged, sorted, grouped, displayed or hidden.

A filter is a way of reproducing (on demand) any useful arrangement of the data that can be created in the upper pane. Each filter can be public (available to all project users) or private (available only to the workstation on which it was created). A filter consists of:

- A list of columns in the order in which they are to appear
- The columns to which a first through fourth-order sort will be applied
- A query to be applied to the data (the query can both AND and OR conditions)
- A context for the query. For example, the query can be applied to all the files in the view, only the files that are not in the view, or both types of files so long as they have not been excluded from the project and its views.

For more information about filters, etc. see "Managing Data in the Upper Pane" on page 53.

To select or create filters, queries, and sort orders, right-click on a column header for a to display a context menu that enables you to perform these operations.

/To sort the displayed files based on the value in that column, click a column header. The sort is in ascending numeric, alphanumeric, or internal key order, depending on the data. To change the sort order from ascending to descending (or vice versa), click the header a second time. The application displays a triangle in the column header. It points up (ascending order) or down (descending order).

There are several ways to refresh the project view window or the upper pane.

- Pressing *F5* refreshes the upper right pane for the current item
- Pressing *Ctrl+F5* refreshes the upper pane and simultaneously collapse all open groups

- Pressing *Shift + F5* refreshes the entire view (all item lists in all the tabs as well as the folder hierarchy)
- Pressing F6 refreshes the files from an external archive, such as Visual SourceSafe or PVCS. Be sure to enable the All Descendants button (or select All Descendants from the File menu) before you press F6 so that the refresh recurses through all the child folders.

Using the Lower Pane

The lower pane displays information about the item selected from the upper pane. The information that appears depends upon the tab you select .

Selecting the Detail Tab

Selecting the Detail tab lists vertically the information that is displayed horizontally in the upper pane for the selected item.

The following section of the project view window shows some properties of the file Stardoc.ico. The columns in the upper pane are Name, Status, Locked By, Vault Branch, and so on. These same properties appear in the Property column in the lower pane. The value of each property is beside its name.

Even when tasks and topics are displayed as trees in the upper pane, the Display pane lists the fields that would have been displayed in the upper pane using the list format.

The Display pane has no context menu.

Selecting the History Tab

Selecting the History tab lists the past revisions for the selected item (unless it is an audit entry or a file whose status is Not In View or Unknown, in which case the tab displays nothing at all).

The revisions are listed from the tip revision to the initial revision. A red arrow indicates the revision on disk (the revision in the repository that you added, checked in, or checked out most recently). The red arrow applies only to files.

Double-clicking a revision displays its properties.

Right-clicking displays a context menu that enables you to review the revisions properties, to compare revisions, to change the comment associated with that revision, and to see its references. For files, you can also view the file in the default editor (usually Notepad) or open in an appropriate application.

Selecting the Label Tab

Selecting the Label tab displays a tree of the labels attached to each revision of the selected item. You can use drag-and-drop to move a label from one revision to another.

A label’s type (view or revision) precedes its name in the tree.

Right-clicking a revision displays a context menu from which you can attach and detach a label, etc.

Selecting the Link Tab

Selecting the Link tab lists the items to which the selected item has been linked. For example, a file might be linked to a folder, another file, a change request, a task, or a topic. (Audit entries cannot be linked, so the lower pane is empty when you select an audit entry.)

Right-clicking a link displays a context menu from which you can see the properties of either the link or the item listed in the link list. You can also delete the link.

Selecting the Reference Tab

Selecting the Reference tab displays the selected item's references. Because of sharing and because views are children of other views, a folder or item can be associated with more than one project, view, or parent folder (as long as they all use the same server configuration). Each instance of the folder or item has a reference to it.

For folder references, you select Folder > Advanced > References; for item references, you use the project view window and the Reference tab on the lower pane.

The Current icon indicates the reference that is currently selected. Otherwise, this pane contains the same information regardless of the view in which you select the item.

Each reference has several parts including the project name, the view path, the folder path, and the branch revision.

The references in bold indicate which revisions of the folder or item are its descendants. In other words, the selected folder or item is part of the history and the revisions that are in bold.

The Reference pane has no context menu.

Using the Title Bars

The title bars provide information about the project and view. The title bars display:

- Product name.

- Version number (follows the product name in the Windows client only).
- Server description (the name you provided for the server configuration that contains the currently displayed project view).

Note

If you are using a Windows client, you may see the name of the computer on which the Server is installed instead of a server description.

You may have manually opened the first window with a shortcut, or the application may have automatically opened the window with a shortcut. If the Restore Workspace On Startup option is selected and the window was open when you exited your last session, the application opens the window automatically.

- Project name (StarDraw)
- View name (StarDraw)
- View's working folder (C:\StarDraw)
- Current Profile (Default). Because the Cross-Platform client has no profiles, its title bar always displays the word Default.

Using the Status Bar

The status bar at the bottom of the project view window displays the following information:

- The configuration for the view. For example, it may be:
  - Current (the default view configuration)
  - Rolled back to a label (for example, View - Label: Build 6)
  - Rolled back to a promotion state (for example, View - Promo: Development)
  - Rolled back to a specific time (for example, View - As of: 3/11/03 18:55:02)

  If the view is read-only (i.e., it cannot be changed), R/O appears before the configuration information. All rolled-back views are read-only because you cannot change the past.
- The active process item or the fact that there is no active process item
- The number of items in the upper pane and the number of items selected from the pane
- Whether a query (other than the one that is part of the filter) has been applied to the upper pane
- The name of the logged-on user
- StarTeamMPX information

  A lightening bolt changes color or is crossed out to let you know if StarTeamMPX is available, available but disabled, or available and disconnected. The words Instant, Auto, and Manual provide refresh information. For more details, see "Using StarTeamMPX" on page 282.

- The Cross-Platform client displays:
  - The same notification icons that are found in the Windows system tray if you are using the Windows client. See “Controlling System Tray Notification” on page 281 for more information about this type of notification.
  - The operating system locale and the flag for the locale. For example, for US English, you see EN and the American flag.
  - The Server’s build number.

Using Version Control

Version control applications store past revisions of files (primarily or even solely source code text files). For example, you can return to an earlier revision of a file to reclaim source code that should not have been removed. The application can include any type of product-related files, including source code files, marketing documents, help files, user documentation, test suites, functional specifications and diagrams.

Version control for files is only one aspect of the application’s version control system. The application also stores all revisions of other types of items, such as change requests, and also provides useful historical information. Even though the application provides a greater number of version control features for files, it provides useful historical information for other components.

However, the application provides the greatest number of version control features for files. When you are working with files, the application does the following:

- Tracks files that need to be checked in or out and enables you to check files in or out from several folders in a single operation.
- Allows you to easily review the history of a file and compare its past revisions.
- Automatically merges changes made by two users who have modified the same file. This merge is done by using the Visual Merge utility.
- Supports branching supported through views.
- Allows you to establish promotion states, through view labels, enabling groups of files to pass from one stage to another in the production process. For example, promotion states might include Development, Test, and Release states.
- Permits keyword expansion, which can embed version control information into text files for quick and convenient reference.
- Provides configurable version control. For example, when a file is checked out that is not locked, you can require the working copy to be made read-only.

The History tab plays an important role in version control. After you select the appropriate file from the upper pane, select the History tab to display a list of all the past revisions of that file. Using the History pane’s context menu, you can check out and compare revisions. The following illustration of the project view window shows a selected file and its history. It also shows the History pane’s context menu.

For more on files, see “Managing Files” on page 127. For more about change request, task, and topic revisions, see “Comparing Properties” on page 203.

Using Defect Tracking

A defect tracking system enables users to track problems with and suggestions about a project or product. Like other such system, the application allows you to enter such defects and assign them to a team member, who is responsible for fixing it. The defect is then fixed or resolved in another way. For example, a defect might not be fixed because:

- It is a duplicate of another defect.
- The team member trying to resolve it cannot reproduce it.
- It is deferred until a later release.
- It is as designed.

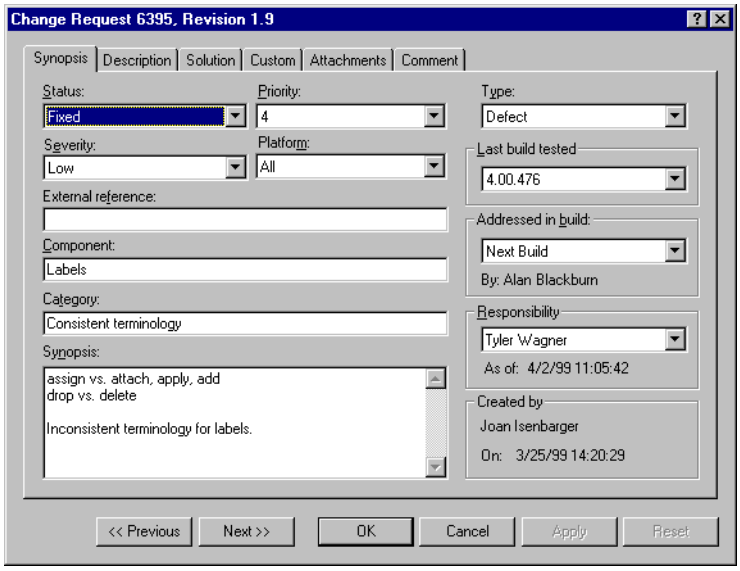
For more information on the application's defect tracking system, see "Tracking Change Requests" on page 155.

Again, like other defect tracking systems, the application has search and/or reporting mechanisms so that you can check the defects that have been assigned to you.

However, the application's Change Request component also offers the following additional features:

- Testers can place change requests in specific folders.
- Developers and testers can link change requests to other items, such as files or other change requests.
- Team members can mark defects as fixed at the same time that they check in files. Thus, they can perform one operation in one application, rather than the usual two operations (each in a separate application).
- The fixed defect automatically becomes associated with the next build label. This feature lets testers know exactly what build to test for the fix, avoiding confusion about when the fix was made in the product.
- Team members can enable system tray notification while running the application, to check for change requests that have been sent to them; see "Controlling System Tray Notification" on page 208.
- If the administrator has enabled e-mail notification, a team member will automatically receive an e-mail message if:
  - The team member is assigned the responsibility for a change request.
  - The team member is responsible for a requirement or task, and one of the fields of the requirement or task changes.
  - The team member is listed as a recipient for a topic, and any field for that topic changes.

The following figure shows a change request for a defect:

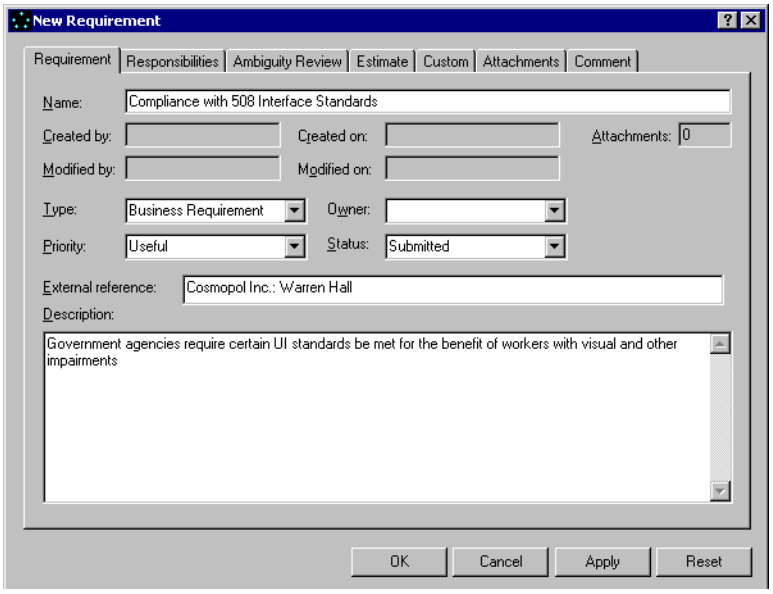


For more information about change requests, see “Tracking Change Requests” on page 155.

Using Requirements

The Requirement component enables team members to create requirements in the application. Requirements can also be imported from CaliberRM.

The following figure shows a typical requirement:



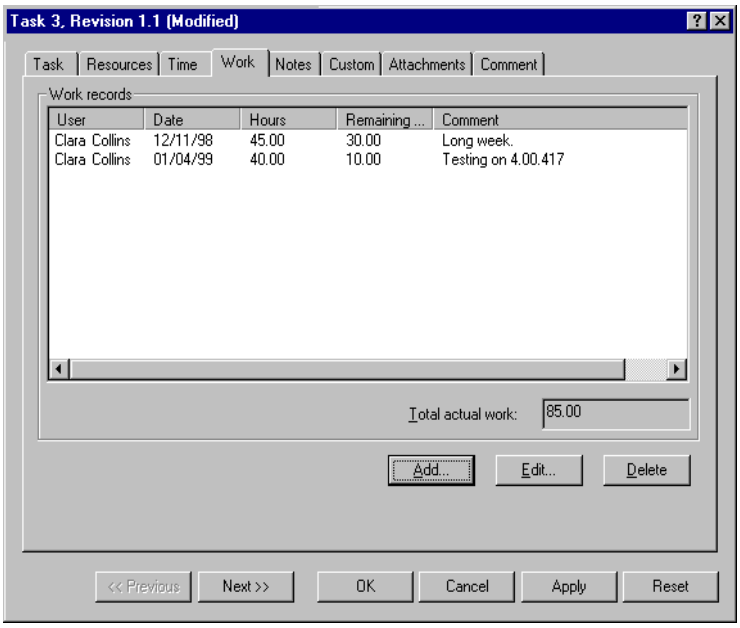
Organizations that use the application can implement requirement-driven development processes. Users can manually link the files that complete requirements to those requirements or use requirements as process items so that the linking is done automatically.

Using Tasks

The Task component enables team members to create task lists and work assignments in the application. Projects from MS Project can be exported to the application or the Task component can be used as a stand-alone.

Tasks enable users to define their work and report their efforts. In combination with links, developers can associate tasks with change requests, files and topics. Tasks are displayed in either a tree format or list format. The tree format enables team members to see the relationship between tasks and their subtasks. This tree of tasks and subtasks is called the task hierarchy. The list format permits team members to sort, group, query and select specific fields to display. Task icons are provided to identify a task’s status, priority, milestone and need for attention.

The following figure shows the number of hours of work logged for a particular task:



For more information about tasks, see “Managing Tasks” on page 171.

Using Topics

By using the Topics component, team members can discuss issues about the project in general, specific files, specific defects, and so on. The historical value of these comments is significant to the project because future team members can:

- Be brought up to date quickly by reading the topics associated with the pieces of the project that are important to them.
- Avoid retrying solutions that were previously found faulty.
- Understand why a particular solution to a problem became necessary and reassess this decision.

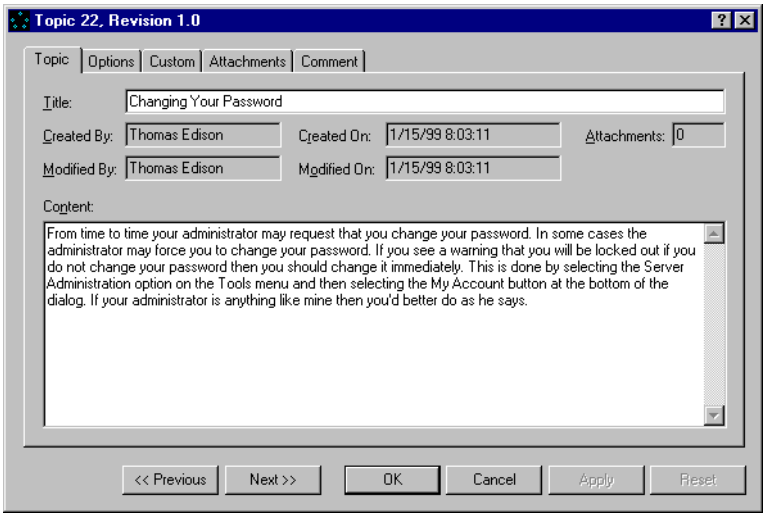
Threaded conversation applications are usually used within companies to organize discussions on product development and similar topics. These applications provide a hierarchical structure for the discussions that consist of electronic messages.

Each topic becomes the root of a tree. Its branches are responses to the topic. Responses to responses form additional branches. Using topics keeps all the messages on one topic together and provide a visual way to see the order and relationship among the responses.

The Topic component not only improves teamwork on product development, but it tightly integrates file version control with its other components. A topic can be assigned to a project view or folder and directly linked to other items.

- A team member can easily incorporate the input from others or ask questions while working on a file.
- Notes explaining why a particular method was or was not used can be included as linked topics.
- Topics can note aspects of the product that may have to be changed in a later release.

The following figure shows a sample topic:



For more information about topics, see “Managing Topics” on page 181

Chapter 3

Using Projects and Views

A project is a way to group related items (such as files and change requests) hierarchically. Views and folders enable you to organize these related items more efficiently. For example, if you create a project for a software product, the files containing the product's functional specification, marketing requirements document, source code, and test suites can each be stored in separate folders.

Views can be used in a variety of ways. For example, different views can be used so that developers see only the project's source code folder and its child folders, marketing personnel see only the project's marketing folder and its child folders, and so on. In this case, each view has a different folder as its root. Views also support branching and parallel development (see "Understanding Branching" on page 197 for more information). At the view level or item by item, you can branch data such as files and change requests. The branching enables you to create a special variation of your product. For example, you can start on the 2.0 version of your product without hampering the creation of service packs for the 1.0 version.

Adding Access to a Server

You can access one or more Servers from the application. You must identify each server by its location or IP address and select an appropriate protocol and endpoint. This will be described in the following steps. After the server is added, users can access whatever projects are available for the current server configuration of that server.

Administrators add server access when creating a project. Team members can add server access as part of opening an existing project.

To add server access:

1. Click the Open Project icon from the toolbar. The *Open Project Wizard* dialog appears.
2. Click Add Server to add a new server.

   The *Add Server* dialog appears.

Important See your administrator for the server address, protocol, and endpoint information. Your administrator can also tell you what MPX profile to use if your server uses StarTeamMPX.

3 Enter an easy-to-remember description in the *Server Description* text box. This is a unique descriptive name for the server. It is *not* case-sensitive and cannot contain colons (:).

4 Enter or browse for the address in the *Server Address* text box. This address is the computer name or IP address.

5 Enter the endpoint in the TCP/IP *Endpoint* text box. The endpoint is the port number.

6 Select the Compress Transferred Data check box if you want to use compression.

7 Select an encryption type check box to encrypt data transferred between your workstation and the server. Encryption protects files and other project information from being read by unauthorized parties over unsecured network lines.

The encryption types are ordered (top to bottom) based on speed. Each encryption type is slower, but safer, than the type that precedes it.

For more information about encrypting and compressing transferred data, see "Transferring Data between Your Workstation and the Server" on page 21.

8 (Optional) If you are using StarTeamMPX and do not want to use the default profile (usually Unicast On-site), click MPX Profiles.

This action displays the *MPX Message Broker Profiles* dialog.

a Select a profile.

b (Optional) Click Properties to review the selected profile's properties.

c Click Set to use the selected profile as your StarTeamMPX profile.

d Click Set.

e Click Close to return to the *Add Server* dialog.

Tip To review a profile's properties, select the profile and click Properties.

To return to the default profile set by the administrator:

1 Click Restore Default.

2 Click OK.

If you were opening a project, the *Open Project Wizard* dialog reopens. From here, follow the steps in the procedure for opening projects.

Opening an Existing Project

Opening a project opens a window within the application and displays the project view that you selected.

Note Before you can open a project, you must have access to the server on which the project resides. See "Adding Access to a Server" on page 19.

Depending on their job responsibilities, some users may have several projects or several views of the same project open simultaneously.

Sometimes a user opens two windows of the same project view. For example, by using two windows for the same project view, you can look at both files and change requests without having to switch tabs. Or you can look at the files from two widely separated folders without having to use the All Descendants button and scroll between the files in each folder.

To open a project window:

1 Do one of the following:

- Click the Open Project icon from the toolbar.

- Select Project > Open from the menu bar. The *Open Project Wizard* dialog appears.

2 Click the plus sign in front of the server name or double-click the server name where the project is located.

If you have already logged on to this server configuration, skip to step 4.

If you have not yet logged on, the *Log On To* dialog appears.

3 Enter your user name and password in the appropriate text boxes.

Usually people use their network log on name as the user name. However, the name and password must be known to the server, and you must have the necessary access rights to continue. Passwords are case-sensitive and may have length restrictions. See your administrator for your server logon name and other details.

Tip If you have already logged onto the server or the Toolbar, but wish to log on as a different user, click Log On As. If you double-click the server name, the user name for your most recent, current logon appears in parentheses, after the name of the server. If you are logging onto this server for the first time in this instance of the application, but have the Toolbar running, the user name in parentheses is the one recognized by Toolbar as your default set of credentials. The “Save as default credentials for this server” check box on the *Log On* dialog resets your default credentials for this server to the user name and password you have just entered in the text boxes of the *Log On* dialog.

Entering your user name and password returns you to the *Open Project Wizard* dialog, which displays a list of projects for that server.

4 Do one of the following:

- Select the project name, then click *Finish* to open your project.
- Double-click the project name to select a specific view of that project.
- Select the project name, then click Next to select a specific view of that project.

5 If the *Open Project Wizard: Select View* dialog appears, select the view name from the View list and click Finish to open your project in that view.

Note When the view icon is greyed-out, you do not have access to that view.

Transferring Data between Your Workstation and the Server

You can encrypt and compress data that is transferred between your workstation and the server. Encryption protects files and other project information from being read by unauthorized parties over unsecured network lines. Compression reduces the amount of traffic on the network. However, the time it takes to compress and decompress the data is added to the transfer time.

As you log onto a server, the encryption and compression properties for the data transferred between your workstation and the server become set and locked. The lock is released when you become disconnected from the server. For example, the server might be shut down or you might close the last project or view that accessed the server. The next time you log onto the server, the encryption and compression properties are reset.

For example, suppose you open the application. You then access server x for the first time during this session by creating Project 25. As part of the creation process, you select encryption and compression properties. Those properties become the properties for data transferred between your workstation and server x.

During the same session, if you open another project or *view* that resides on server x, the application ignores any encryption and compression changes you make as part of

the open-project process. For more information about setting these properties as you open a project, see “Opening an Existing Project” on page 20.

Using View Shortcuts

If you will be accessing a project view frequently, you may want to save the view as a shortcut on your desktop. Double-clicking the shortcut both starts the application and opens the view associated with the shortcut.

To save a shortcut to your project view (along with a particular configuration):

1 Do one of the following:

- Select Project > Save Shortcut As from the menu bar.
- Click Save Shortcut on the toolbar.

The *Save As* dialog appears.

2 Enter a name or use the default name for the shortcut in the *File Name* text box. Keep the .stx extension.

3 Select a location, usually your desktop, to store this view shortcut.

4 Click Save.

To open your project view (to a particular configuration) with a shortcut:

1 Select Project > Open Shortcut from the menu bar.

The *Open* dialog appears.

2 Select the shortcut name and click Open.

Switching Views

You usually select a view as you open the project window. However, you can change what view of that project is displayed in the window at any time. The newly selected view always opens using the current configuration, regardless of the configuration it was in when you last exited it.

To go directly to a specific view and configuration, create a shortcut. See “Using View Shortcuts” on page 22.

To select a different view:

1 Select View > Select View from the menu bar.

The *Select a View* dialog appears and the views are displayed hierarchically.

2 Select the view to be opened.

Important If you want two view windows open at the same time, select Project > Open to display the other view.

Configuring a View to a Point in the Past

You can roll back a view to a past state based on a label, promotion state, or a point in time. For example, you might want to:

- Take a quick look at how things were when the Beta3 label was applied
- Recover an item that has been deleted by rolling back the view to a date before the item was deleted

However, doing this “freezes” the view until you change its configuration back to current or close the project, which automatically changes the configuration back to current. You cannot check in files, update change requests, and so on because you cannot change the past.

You may need to open two instances of the application. For example, suppose you need to examine files in the White Box Test promotion state and enter change requests about problems you find in the files. After you configure the view to the promotion state, it becomes read-only. You need another instance of the view using the current configuration so that you can enter the change requests.

The status bar displays the view’s configuration. It can be:

- Current
- Rolled back to a label (for example, View - Label: Build 6)
- Rolled back to a promotion state (for example, View - Promo: Development)
- Rolled back to a specific time (for example, View - As of: 3/11/99 18:55:02)

To roll back the current view:

1. Select View > Select Configuration from the menu bar. The *Select a View Configuration* dialog appears.
2. Select a view configuration option:
   - Labeled Configuration

     This limits the view to items with the view label that you specify.

     If the view is a branching view, the label must be one created in this view. Reference views can inherit labels from their parent views.

     The existing labels are listed in reverse chronological order based on the time for which they were created.

     This option is disabled if this view has no labels defined for it.
   - Promotion State Configuration

     As you configure a view, you can select a promotion state.

     If the promotion state assigned is <current> rather than a view label, this configuration is still read-only.
   - Configuration As Of

     This rolls back the view to a specific point in time. It includes everything in the view just prior to the specified date and time.

     This option defaults to the current date and time, but you can select one in the past. Depending on the type of view, this time might have to be after the view was created.

To return to the current configuration:

1. Select *View* > Select Configuration from the menu bar.

   The *Select a View Configuration* dialog appears.
2. Select the Current Configuration option button.

Caution Suppose you roll a view back to a label or promotion state. Be aware that any item with that label or the label associated with that promotion state becomes a special case if all of the following are true:

- The item existed at the time that the label was created
- The item still has the label, even though it may now be attached to a different revision
- The item has been moved to a different folder

The rolled-back view displays the item in its original folder, regardless of the item's new location.

An item that was created after the label was created appears in its most recent location in the rolled-back view. An item that no longer has that label, regardless of when the item was created, does *not* appear in the rolled-back view.

Controlling File Status Information

File status information about the files you are working on is stored on your workstation either is a central location or in a child folder (named .sbas) of each working folder.

You control how and where that information is stored by:

- Setting personal options that control file status information for all your files—unless those files are in views for which you have set the view property for file status. See "Setting File Options" on page 275 for more information.
- Setting the file status property for a specific view. The view property defaults to the storage method that you selected as a personal option. When changed from that default, the view property take precedence over your personal option for the view.

To set the file status property for a view:

1. Select View > Properties from the menu bar. The *View Properties* dialog appears.
2. Select the Info tab.
3. In the File Status Repository group box, select the Central or Per Folder option button. The per-folder option is most useful in the special case where multiple users are sharing a working folder; for example, on a shared network drive.

   Suppose several users all check files in and out of StarTeam from a shared working folder, such as \\OurDevServer\SourceCode\WorkingFolder <file:///\\OurDevServer\SourceCode\WorkingFolderr>. If these users have set the central repository option for file statuses, the statuses are stored on each of their computers. Whenever a user makes a change to a file in the working folder, the status for that file is undated only on that user's computer. Everyone else sees the status "Unknown" for that file. Over time, all the files may have been changed, and the statuses can become "Unknown" for all users of all files. Using the per-folder option causes the statuses to be updated within the working folder itself. Everyone has access to those status changes and "Unknown" statuses do not occur.
4. Click OK.

Tip You can return to using the personal options by selecting the Default option button.

Using Alternate Working Folders

The view's working folder may not be the optimal choice for all users. You, or any other user with the access rights to do so, can select a more useful location for the view's working folder on your own workstation by designating an alternate working folder. For example, you might want to use a shorter path or a different drive letter. Remember that a working folder must point to a physically discrete location, such as a drive on your workstation or a personal directory on a shared file server. Borland does not recommend putting your settings on a mapped network drive.

The alternate working folder path for the view is specific to the workstation and user. For example, if you log onto the project as another user or use another workstation, your alternate working folder setting is *not* known.

When you designate an alternate working folder for the view, the path to the working folder for each child folder in the view may be similarly modified for your workstation.

For every folder in the hierarchy whose working folder is relative to the path of the view's working folder (as opposed to having an absolute path or an alternate working folder path of its own), your alternate path for the view's working folder becomes part of the paths to its child folders' working folders.

To use an alternate working folder for a view:

1. Select View > Properties from the menu bar. The *View Properties* dialog appears.
2. Select the Info tab.
3. In the *Working Folder* group box, select the Alternate option button.
4. Enter or browse for an alternate working folder.
5. Click OK.

The *View Properties* dialog closes, and the specified alternate working folder is used. Files affected by this change in working folders will have the status of Missing, and must be checked out to generate a new set of working files.

To check out all the Missing files:

1. Select the view's root folder.
2. Select the File > Check Out All command.
3. In the resulting *Check Out All Descendants* dialog, make any desired changes, and click OK.
4. In the resulting confirmation dialog, click OK.

You will see a progress dialog while the files are copied from the repository to your working folders. (Empty folders without files in the project are not created in your working folders.)

Using View Profiles

A view profile is a workstation-dependent set of limitations applied to a view. You can create as many profiles for use on your workstation as you wish.

A profile specifies which folders are visible and which EOL and path case sensitivity settings apply to those folders. For example, if you are building a UNIX version and a Windows version of a product from the same view, you may need a profile for each. UNIX expects paths to be case sensitive and uses a linefeed as the EOL (end-of-line) character in a text file, while Windows paths are not case sensitive and Windows text files use the carriage return/linefeed combination to indicate the end of a line.

Every view has a minimum of one profile. That profile is named Default, but new profiles can be created with default settings or based on previous profiles.

Any changes made to a folder's visibility, EOL, and path case sensitivity settings automatically become part of the currently selected profile.

The name of the current profile is displayed near the end of the title bar.

Controlling Folders in Profiles

You can add or remove application folders from a profile by making them visible or invisible.

When a folder is invisible in a given profile, the folder's name remains in the folder hierarchy, but the color of its icon changes from yellow to white, and its contents are no longer displayed.

To change a folder's visibility:

1. Make sure that the profile you want to change is the current profile:
   a. Select View > Profiles from the menu bar.

      The *View Profiles* dialog appears.

   b. Select the correct profile, click Select.

   c. Click Close.
2. Select the folder that you want to make invisible or return to visibility.
3. Select Folder > Properties from the menu bar.

   The *Folder Properties* dialog appears.
4. Select or clear the Visible check box.
5. Click OK.

Controlling EOL Characters and Path Case Sensitivity in Profiles

One profile differs from another based on which folders are visible and on which EOL and path case sensitivity settings are active for those folders.

1. Make sure that the profile you want to change is the current profile:
   a. Select View > Profiles from the menu bar. The *View Profiles* dialog appears.

   b. Select the correct profile, and click Select.

   c. Click Close.
2. Select the folder that requires changes to its EOL and/or path case sensitivity settings.
3. Select Folder > Properties from the menu bar.

   The *Folder Properties* dialog appears.
4. Select the File tab.
5. (Optional) In the EOL Conversion group box, select the Override For This Folder And All Its Child Folders check box to change this setting for the folder's children as well.
6. (Optional) Select the option button for the appropriate EOL character.
7. (Optional) In the *File Path Conventions* group box, select the Override For This Folder And All Its Child Folders check box to change this setting for the folder's children as well.
8. (Optional) Select the Case Sensitive File Paths check box.
9. Click OK.

Reviewing View Properties

Sometimes you may want to look at the values and properties originally used to create a view. Reviewing this information may help you understand the behavior of changes within the view or the views that have been derived from it.

To review the current view properties:

1. Select View > Properties from the menu bar. The *View Properties* dialog appears.
2. Select the Info tab to see:
   - Name and description of the view
   - Who created the view and when it was created

- Whether the items are set to branch on change
- Whether a central or a per folder repository is being used
- Working folder path

Note: Depending upon your access rights, you may be able to change some of these items.

3 Select the Hierarchy tab to review the hierarchy of views for this project.

4 Select the Type tab to see:

- View type
- Whether the view is a branch or a root view
- (For branching views) Whether the original default was Branch All or Branch None
- Parent view on which this view was based
- Parent configuration used to create this view

5 To exit, click Cancel.

Refreshing Views

You can refresh application data by:

- Pressing *F5* to refresh the upper right pane for the current item
- Pressing *Ctrl+F5* to refresh the upper pane and simultaneously collapse all open groups
- Pressing *Shift + F5* to refresh the entire view (all item lists in all the tabs as well as the folder hierarchy)
- Pressing *F6* to refresh the files from an external archive, such as Visual SourceSafe or PVCS. Be sure to enable the All Descendants button (or select All Descendants from the File menu) before you press *F6* so that the refresh recurses through all the child folders.

Reviewing Connection Properties

From the application, you can open several views from one or more projects. You can access projects from one or more server configurations. Sometimes your company will use the same project names on different servers or the same view names within different projects.

Reviewing the connection properties lets you verify the server, address, end point (port), logged on user, and logon time.

To check connection properties:

1 Select Project > Connection Properties from the menu bar. The *Connection Properties* dialog appears.

2 Click OK.

Logging Off the Application

When you close the application or the last window that displays a project view for a given server with a given user name and password, you log off the server. The Cross-Platform client provides a special Log Off command that enables you to close all those

windows simultaneously. This operation also performs some clean up operations required only by those clients.

To close all the windows to a given server using a given user name:

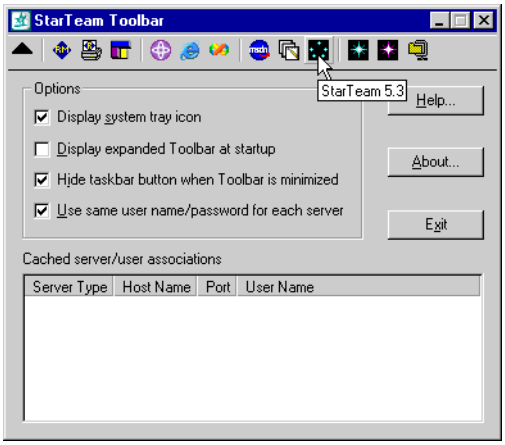
1. Select Project > Log Off from the menu bar.
2. In the resulting dialog, click Yes to confirm.

Chapter:

4

Using the Toolbar

The Toolbar is a component of the StarTeam and CaliberRM products that is designed to make it easier for you to log onto multiple servers and to launch different programs.



The Toolbar automatically caches the user name and password used to log onto each StarTeam or CaliberRM server, reducing the number of times you must log onto the same server to perform certain operations. (You can override the automatic log on feature when necessary. For details, see “Logging On As a Different User” on page 34.)

The Toolbar is initially populated with shortcuts for the tools of the products that are installed on your workstation. Because the Toolbar uses the standard Windows program shortcut feature, you can easily add any other program as a tool.

The Toolbar runs only on the Windows family of operating systems, and it is installed by default as part of the Windows client and the Cross-Platform client when it is installed on Windows.

Note This chapter describes the behavior of the Toolbar while running on Windows NT Workstation, version 4.0. Due to variations in the different editions of Windows, you may notice some differences on your computer..

The Toolbar enables you to:

- Specify which portions of its user interface are visible (none, system tray only, window only, or system tray and window). By default, only the Toolbar system tray icon is displayed.
- View the cached server/user associations.
- Use the same user name and password for each server, so you do not have to log onto other servers where that user name and password is valid.
- Add program shortcuts as tools on the Toolbar, so you can easily start those tools. (You can also modify or delete an existing tool.)
- Display the Toolbar help information or *About* dialog.

Starting the Toolbar

The Toolbar is started automatically by the clients. To manually start the Toolbar, select Start > Programs > StarTeam > StarTeam Toolbar.

When the Toolbar is started for the first time, only its system tray icon displays.

The Toolbar is initially populated with shortcuts for the tools of the products that are installed on your workstation. These tools are present on the popup menu of the system tray icon, and on the toolbar region of the window.

To make the Toolbar more useful, you can:

- Change the values of its options. (See “Specifying the Toolbar User Interface” on page 31 and “Logging Onto Servers” on page 33.)
- Add, modify, or delete tools. (See “Customizing the Toolbar” on page 35.)

Opening the Toolbar Window

You can open the Toolbar window by choosing the Open command from the context menu of the Toolbar system tray icon.

When no portion of the Toolbar user interface is visible, you can display its expanded window by starting a second instance of the Toolbar. (To do so, select Start > Programs > StarTeam > StarTeam Toolbar.)

Specifying the Toolbar User Interface

The Toolbar has options that enable you to specify which portions of its user interface are visible:

- None
- System tray only
- Window only (either collapsed or expanded)
- System tray and window

Toolbar functionality varies according to the level of its visible user interface.

- Even when no user interface is visible, the Toolbar can cache the user name and password used to log onto each server. (This happens when the “Save as default credentials for this server” option is selected in the *Log On* dialog.)

  When the “Use same user name/ password for each server” option is selected, the first saved user name and password is automatically used for each additional server, so that you do not have to log onto other servers for which that user name and password is valid.
- When the system tray interface is visible, you can also:
  - Open the Toolbar window.
  - Launch an existing tool.
  - Display the Toolbar help information or *About* dialog.
  - Exit the Toolbar.
- When the window is visible and is in its collapsed state, you can also:
  - Add other program shortcuts as tools on the Toolbar, so you can easily locate and start those tools. (You can also modify or delete an existing tool.)
  - Expand the window, to access additional functionality.
- When the window is visible and is in its expanded state, you can also:
  - Set the Toolbar options.
  - View the cached server/user associations.

Displaying the Toolbar Window at Startup

The Toolbar provides a window that can be collapsed or expanded and which has a context menu.

The visibility of this portion of the user interface is determined by the value of the “Display expanded Toolbar at startup” option:

- When this option is selected, the expanded Toolbar window is automatically opened when the Toolbar is started.
- When this option is cleared, the Toolbar window is not displayed when it is started. However, you can open the window by choosing the Open command from the context menu of the system tray icon.

Note When no portion of the Toolbar user interface is visible, you can display its expanded window by starting a second instance of the Toolbar. To do so, select Start > Programs > StarTeam > StarTeam Toolbar.

The context menu for the toolbar area of the Toolbar window provides the following commands:

- **Add Tool**, which opens a wizard that enables you to create a standard Windows shortcut in the Tools folder for the Toolbar. (For details, see “Adding a Tool” on page 35.)

  After adding or modifying a tool, you must choose the Refresh Tools command for the changes to be reflected in the toolbar region of the window and on the context menu for the system tray icon.
- **Modify Tool**, which opens the *Properties* dialog for the selected tool’s program shortcut. (For details, see “Modifying the Properties of a Tool” on page 36.)
- **Delete Tool**, which deletes the selected tool’s program shortcut from the Tools folder, deletes the tool’s entry from the context menu for the system tray icon, and refreshes the toolbar region of the window to delete the tool’s icon. (For details, see “Deleting a Tool” on page 37.)
- **Refresh Tools**, which refreshes the toolbar region of the window and the context menu for the system tray icon, by processing all the program shortcuts in the Tools folder. (For details, see “Refreshing the Toolbar” on page 37.)
- **Open Tools Folder**, which opens an Explorer window for the folder containing the program shortcuts used by the Toolbar.
- **Help**, which displays the Toolbar help information. (For details, see “Displaying the Toolbar Help Information” on page 38.)
- **About**, which displays the Toolbar *About* dialog. (For details, see “Displaying the Toolbar About Dialog” on page 39.)
- **Exit**, which exits the Toolbar. (For details, see “Exiting the Toolbar” on page 39.)

Hiding the Taskbar Button While the Toolbar Window is Minimized

When the “Hide taskbar button when Toolbar is minimized” option is selected, this button does not appear on the Windows taskbar when the Toolbar window is minimized.

When this option is cleared, this button is included on the Windows taskbar when the Toolbar window is minimized.

Hiding the Toolbar System Tray Icon

The Toolbar provides a system tray icon, which has a context menu.

The visibility of this portion of the user interface is determined by the value of the “Display system tray icon” option:

- When the “Display system tray icon” option is selected, an icon for the Toolbar is displayed in the Windows system tray.
- When this option is cleared, no icon for the Toolbar is displayed in the Windows system tray.

The context menu for the Toolbar system tray icon provides the following commands:

- **Open**, which opens the Toolbar window. (For details, see “Opening the Toolbar Window” on page 30.)
- **<tool>**, which launches that tool. (For details, see “Launching a Tool” on page 35.)
- **Help**, which displays the Toolbar help information. (For details, see “Displaying the Toolbar Help Information” on page 38.)
- **About**, which displays the Toolbar *About* dialog. (For details, see “Displaying the Toolbar About Dialog” on page 39.)
- **Exit**, which exits the Toolbar. (For details, see “Exiting the Toolbar” on page 39.)

Hiding the Toolbar User Interface

If you want to use the Toolbar to automatically cache the user name and password you use to log onto each server (or to use the same user name and password for each server), but do not want to launch tools from the Toolbar, you can hide its user interface. To do so, clear both the “Display system tray icon” and the “Display expanded Toolbar at startup” options.

When no portion of the Toolbar user interface is visible, you can display its expanded window by starting a second instance of the Toolbar. To do so, select Start > Programs > StarTeam > StarTeam Toolbar.

Logging Onto Servers

A client maintains a server connection for a specific project view during the client session. Prior to the 5.3 release, if you wanted to access the same project view from a different client, you would have been required to log on again.

The Toolbar now caches a default credential for each server you log onto (unless you clear the “Save as default credentials for this server” option in the *Log On* dialog), so that you do not have to log onto that server again from any client during the current Toolbar session.

If you need to log onto that server as a different user at a later time, you can override the automatic logon feature by clicking the Log On As button in the *Open Project Wizard*. For details, see “Logging On As a Different User” on page 34.

When the “Use same user name/ password for each server” option is selected, you do not even have to log onto other servers where that same user name and password is valid. (On a server where that user name and password is not valid, you can log on as a different user.)

Viewing your Cached Server/User Associations

When you log onto a server and the “Save as default credentials for this server” option is selected in the *Log On* dialog, the Toolbar caches the user name and password you have used to log onto that server. (You can override the automatic log on feature when necessary. For details, see “Logging On As a Different User” on page 34.)

When the “Use same user name/ password for each server” option is cleared, a different user name and password can be saved for each server. The lower portion of the expanded Toolbar window includes a table that displays information about the currently cached server/user associations.

- The **Server Type** column shows the type of server (StarTeam or CaliberRM).
- The **Host Name** column shows the network name of the machine where the server is located.
- The **Port** column shows the port number of a particular server configuration, so you can distinguish between multiple servers running on the same machine.
- The **User Name** column shows the user name portion of the default credentials saved for that server.

If necessary, you can resize the table’s columns or the entire window, to better display the current information.

When the “Use same user name/ password for each server” option is selected, the first saved user name and password is automatically used for each additional server you try to access, so you do not have to log onto other servers where that user name and password is valid.

Note The server/user associations are cached only during the current session of the Toolbar. When you exit the Toolbar, this cache is cleared.

Logging On As a Different User

If the “Save as default credentials for this server” option is selected (which it is by default) in the *Log On* dialog when you log onto a server, the Toolbar caches the user name and password so you will not have to log onto that server again during the Toolbar current session. The cached server/user associations are shown in a table in the lower portion of the expanded Toolbar window.

When a default user credential has been saved for a particular server, the user name appears in parentheses after that server’s name in the Server/Project tree of the *Open Project Wizard*.

If you later need to log onto that server as a different user, you can override the automatic log on feature by clicking the Log On As button. In the resulting *Log On* dialog, you can enter a different user name and password.

Note that the “Save as default credentials for this server” option is cleared by default. (If you select this option, the new user credential will replace the previously saved one.)

When the “Use same user name/ password for each server” option is selected and you try to access some other server where the default credential is not valid, the automatic log on attempt fails and the *Log On* dialog is displayed, so you can log on to that server as a different user (or with a different password).

If the “Save as default credentials for this server” option is selected in the *Log On* dialog, then the “Use same user name/ password for each server” option is cleared and the new user credential is cached for this server.

Using the Same User Name and Password

When the “Save as default credentials for this server” option is selected in the *Log On* dialog, the Toolbar caches the user name and password used to log onto that server. This reduces the number of times you must log onto the same server to perform certain operations. (You can override the automatic log on feature when necessary. For details, see “Logging On As a Different User” on page 34.)

When the “Use same user name/ password for each server” option is cleared, a different user name and password can be saved for each server. (The currently cached server/user associations can be viewed in the expanded Toolbar window, as explained in “Viewing your Cached Server/User Associations” on page 33.)

When the “Use same user name/ password for each server” option is selected, the first saved user name and password is automatically used for each additional server you try to access, so you do not have to log onto other servers where that user name and password is valid.

Note If the automatic log on attempt fails, the *Log On* dialog is displayed, so you can log onto that server as a different user. If the “Save as default credentials for this server” option is selected in the *Log On* dialog, then the “Use same user name/ password for each server” option is cleared and the new user credential is cached for this server.

Launching a Tool

You can launch a tool on the Toolbar either by:

- Choosing the tool's entry on the context menu for the Toolbar system tray icon
- Clicking the tool's icon on the Toolbar window

Customizing the Toolbar

The Toolbar is initially populated with shortcuts for the tools of the products that are installed on your workstation. If you want to launch other programs from the Toolbar, you must add program shortcuts to it. The tools appear on the Toolbar in alphabetical order, based on the name of their program shortcuts.

You can customize the Toolbar either by:

- Using the commands on the Toolbar window's context menu, which is accessed by right-clicking on the toolbar region.
- Working directly with the program shortcuts in the Toolbar Tools folder, using an Explorer window.

After adding, modifying, or deleting a tool, you must use the Refresh Tools command so the changes will be reflected in the toolbar region of the window and on the context menu for the system tray icon.

Detailed procedures for customizing the Toolbar are provided in the following topics:

- Adding a Tool
- Modifying the Properties of a Tool
- Deleting a Tool
- Refreshing the Toolbar

Adding a Tool

You can add a tool to the Toolbar either by:

- Creating a new program shortcut (using the wizard that results from choosing the Add Tool command)
- Copying an existing program shortcut to the Toolbar "Tools" folder (using Windows Explorer)

After adding a tool to the Toolbar, you must choose the Refresh Tools command to refresh the toolbar region of the window and the context menu for the system tray icon. (You might also want to modify the tool's properties. For details, see "Modifying the Properties of a Tool" on page 36.)

Note If you add so many tools to the Toolbar that they cannot be displayed at the window's current size, you should enlarge the window so all the tool icons are visible.

To add a tool by creating a new shortcut:

1. Right-click on the toolbar region of the Toolbar window, and choose the Add Tool command.
2. On the *Create Shortcut* page of the resulting wizard, specify the command line for starting the tool, and click the Next button.

If you do not know the exact location of the program's executable file, then click the Browse button to open a dialog that enables you to locate and select the necessary file.

3 On the *Select a Title for the Program* page of the wizard, specify the name for the tool's shortcut, and click the Finish button.

The specified program shortcut is created in the Toolbar Tools folder.

4 Right-click on the toolbar region of the Toolbar window, and choose the Refresh Tools command.

The tool's icon is added to the toolbar region of the window, and the shortcut's name is added to the context menu for the system tray icon.

To add a tool by copying an existing shortcut:

1 Right-click on the toolbar region of the Toolbar window, and choose the Open Tools Folder command. The resulting Explorer window shows the contents of that folder, which should consist of the program shortcuts for the tools on the Toolbar.

2 In Windows Explorer, locate an existing program shortcut.
(On Windows NT, search for the "Shortcut" file type. On later editions of Windows, search for files named "*.lnk".)

3 Copy the existing program shortcut to the Toolbar Tools folder.

4 Right-click on the toolbar region of the Toolbar window, and choose the Refresh Tools command.

The tool's icon is added to the toolbar region of the window, and the shortcut's name is added to the context menu for the system tray icon.

Modifying the Properties of a Tool

You might want to modify the properties of a tool on the Toolbar. For example, you might want to rename a tool to include the program's version number. Or you might have accepted the default name while creating the shortcut, and you want to change it from the name of the executable file to something more descriptive.

The program shortcut for a tool on the Toolbar has properties that you can modify by using a *Properties* dialog. That dialog can be opened in two ways:

- Right-click on a tool's icon (in the toolbar region of the Toolbar window), and choose the Modify Tool command.
- Right-click on the toolbar region of the Toolbar window and choose the Open Tools Folder command. In the resulting Explorer window, right-click on the program shortcut for the tool, and choose the Properties command.

You can use the Windows "What's This?" help feature to obtain a brief description of a specific property:

1 Click the button with a question mark icon (in the upper-right corner of the dialog). The cursor changes to include a question mark.

2 Click on the property for which you need help. This action displays a small help window.

After modifying a tool's properties, remember to right-click on the toolbar region of the Toolbar window and choose the Refresh Tools command, so that the Toolbar will process all the program shortcuts in its Tools folder.

Note The *Properties* dialog in Windows NT does not allow you to change the name of the shortcut. However, you can rename a shortcut from an Explorer window (by editing the shortcut name), as explained in the following procedure.

To rename a tool (in Windows NT):

1. Right-click on the toolbar region of the Toolbar window and choose the Open Tools Folder command.
2. In the resulting Explorer window, do either of the following:
   - Right-click on the program shortcut for the tool, and choose the Rename command.
   - Click on the program shortcut for the tool, and click again on the text below the icon.
3. Edit the name, and press the Enter key.
4. Right-click on the toolbar region of the Toolbar window, and choose the Refresh Tools command.

Deleting a Tool

If you need to delete a tool on the Toolbar, you can do so from either the Toolbar window (using its Delete Tool command) or an Explorer window.

To delete a tool from the Toolbar window:

1. Right-click on the tool's icon (in the toolbar region of the Toolbar window), and choose the Delete Tool command.
2. In the resulting confirmation dialog, click Yes. The Toolbar is automatically refreshed.

To delete a tool from an Explorer window:

1. Right-click on the toolbar region of the Toolbar window and choose the Open Tools Folder command.
2. In the resulting Explorer window, do one of the following:
   - Right-click on the shortcut, and choose the Delete command.
   - Click on the shortcut, and press the Delete key.
3. In the resulting confirmation dialog, click Yes.
4. Right-click on the toolbar region of the Toolbar window, and choose the Refresh Tools command.

Refreshing the Toolbar

After adding, modifying, or deleting a shortcut for a tool on the Toolbar, you must refresh the Toolbar so that it reflects your changes. This action is necessary regardless of whether you used the commands on the Toolbar window's context menu, or worked directly with the shortcuts in the Tools folder. (The only exception is deleting a tool using the Delete Tool command, which automatically refreshes the Toolbar.)

To refresh the Toolbar, right-click on the toolbar region of the Toolbar window and choose the Refresh Tools command. This command refreshes the toolbar region of the window and the context menu for the system tray icon, by processing all the program shortcuts in the Tools folder.

Managing the Toolbar Window

This group of topics explains the following operations you can perform on the Toolbar window:

- Collapse or expand it

- Resize it
- Minimize it
- Close it

You can open the Toolbar window by choosing the Open command from the context menu of the Toolbar system tray icon. When no portion of the Toolbar user interface is visible, you can display the expanded window by starting a second instance of the Toolbar. (To do so, select Start > Programs > StarTeam > StarTeam Toolbar.)

Collapsing and Expanding the Toolbar Window

The Toolbar window can be displayed in either an expanded or a collapsed state. To switch between these states, click the leftmost button (with a black triangle as its icon) on the toolbar.

Note that the window must be expanded to set the Toolbar options or to view the cached server/user associations.

Resizing the Toolbar Window

You can resize the Toolbar window in both dimensions while it is expanded, but in only the horizontal dimension while it is collapsed. (The Toolbar has a minimum size for its expanded and collapsed states.) The window's dimensions and location are saved when the Toolbar window is closed.

The columns of the table showing the cached server/user associations can also be resized to better display the current contents. However, these adjustments are not saved when the Toolbar window is closed.

Minimizing the Toolbar Window

When you are not using the Toolbar window, you can minimize it, to reduce the clutter on your Windows desktop.

The "Hide taskbar button when Toolbar is minimized" option determines whether a button is included on the Windows taskbar when the Toolbar window is minimized. (For details, see "Hiding the Taskbar Button While the Toolbar Window is Minimized" on page 32.)

Closing the Toolbar Window

Closing the Toolbar window is equivalent to exiting the program. For details, see "Exiting the Toolbar" on page 39.

Displaying the Toolbar Help Information

You can display the Toolbar help information by any of the following methods:

- Choosing the Help command on the context menu for the system tray icon
- Choosing the Help command on the context menu for the window
- Clicking the Help button on the expanded window

Displaying the Toolbar About Dialog

You can display the Toolbar *About* dialog by any of the following methods:

- Choosing the About command on the context menu for the system tray icon
- Choosing the About command on the context menu for the window
- Clicking the About button on the expanded window

This *About* dialog displays the Toolbar version information and has the following buttons:

- **OK**, which closes this dialog.
- **System Info**, which runs the Microsoft System Information utility.

Exiting the Toolbar

When you exit the Toolbar:

- The cache of server/user associations is cleared.,
- Certain information about the state of the Toolbar is saved in the Windows registry, including the size and location of its window, and the current values of its options.
- The program stops running.

You can exit the Toolbar in various ways, depending on what portion of its user interface is visible:

- When the Toolbar window is visible and is in its expanded state, you can right-click on the toolbar region and choose the Exit command, click the Close button in the upper-right corner of the window, or click the Exit button.

  You can also hold down the *Alt* key and press the *F4 k*ey, or click the icon in the upper left corner of the window and choose the Close command.
- When the Toolbar window is visible and is in its collapsed state, you can right-click on the toolbar region and choose the Exit command, or click the Close button in the upper-right corner of the window.
- When the Toolbar system tray icon is visible, you can right-click on the icon and choose the Exit command.
- When no portion of the Toolbar user interface is visible, you can end the task using the Windows Task Manager, or start a second instance of the Toolbar which will display the expanded window.

To exit the Toolbar (using the Windows Task Manager) when no portion of the user interface is visible:

1. Right-click on an open space of the Windows taskbar, and choose the Task Manager command.
2. In the resulting *Windows Task Manager* dialog:
   a. Select the Processes tab.
   b. Select the SBToolbar.exe entry.
   c. Click the End Process button.

Chapter

5

Managing Folders

The administrator usually creates projects and project views. If you are a typical user, you routinely open a particular project view and manage “your” folders and their contents, such as files and change requests.

Managing application folders is very similar to managing a project. You can create folders, delete folders, and modify their properties—if you have the correct access rights. One of the most important properties to notice about your folder is its working folder. The reason for this is that you will need to know where on your workstation the application will copy file revisions that you check out and where new revisions must be so that you can check them in.

A number of other operations can be performed on folders, such as moving a folder or changing its branching behavior. See “Performing Generic Operations” on page 191 for more information.

Understanding the Folder Hierarchy

A folder is a container. You can group items within a project view by placing them into folders. For example, a folder named Source Code can contain source code files and requested changes to those files. Folders can be created automatically at the time a project is created. They can also be added later by you or others. Projects are usually created by an administrator or team leader, but anyone can create projects if they have the correct access rights. See your administrator if you have any further questions regarding the access rights assigned to you.

The parent or root folder of a project is always created at the same time as the project. It is actually the root folder of the project’s root (or initial) view. The project, view, and this root folder initially have the same name (although those names can be changed).

When users create projects, they select locations on their workstations as the working folders for those projects. The working folder designated for a project also becomes the working folder for the project’s root view and for the root folder in that view’s folder hierarchy.

A working folder is the actual location on your workstation where files are stored while you are using them. A working folder is a property of a folder. Despite the fact that these are both called folders, the working folder and the folder are not identical. Their differentiating characteristics include:

- The path to the working folder can be totally different from the path within the application to the application folder.
- An application folder is an object controlled from within the application.

  The data associated with this folder is stored in the database that stores all the project data.
- A working folder is an object controlled by your operating system. It stores files that are checked out from the application.

Usually, the user who creates a project sets up a hierarchy of folders on a workstation before creating the project. The user designates the root folder of that hierarchy as the project's working folder. Then the application can automatically create an application folder for each of the child folders in the hierarchy. The child folder becomes the application folder's working folder.

If child application folders are created at the time the project is created, then:

- The application folders' working folders were part of an existing hierarchy on the project creator's workstation.
- Their names are the same as the names of their working folders, but they can be changed later.
- Their working folders remain hierarchically connected to the root folder's working folder. That is, if you change the path to the root folder's working folder, you also change the path to this folder (unless you manually set an absolute path for these working folders). In other words, the application stores a relative path to each child folder.

A project, its root view, and the root view's root folder all have the same working folder. For additional views, each view and its root folder have the same working folder.

The working folder for the view/root folder always has an absolute path (starting with the drive letter and specifically naming the folders at subsequent levels until you reach the working folder itself). In this case, that absolute path is C:\StarDraw.

If you look at the root folder's properties, you will see that the working folder is the same. However, it is displayed in the Complete Working Folder Path display box instead of the Default text box. All the text boxes for the root folder's working folder are always disabled. This is because you can only change this working folder at the view level.

For the child folders that were created at the same time as the project, the application stores the path to each working folder as a relative path.

Adding New Folders

Application folders can be added to a project view using Folder > New or dragging a folder from Windows Explorer to the application. When a new folder is added:

- The new application folder's working folder does not have to belong to the same hierarchy as the other application folders' working folders. However, if it uses the same drive letter as the root folder's working folder, its path is stored as a relative path based on the path to the working folder of its parent folder in the hierarchy.
- Its name can be different from the name of its working folder.
- If the new working folder has child folders, a folder can be created for each of the children. Essentially, the newly added folder becomes the root of a new branch of folders.
  - The application folders created for the child folders take the names of their working folders—at least initially.

- The working folders retain their relationship to the working folder that is the root of their hierarchy (that is, the working folder for the newly added folder). If you change the path to the newly added folder's working folder, you also change the path to these working folders (unless you manually set an absolute path for these working folders).

Note The process of adding a folder is explained in detail in "Adding Folders to a View" on page 44. This section explains new folders in terms of their relationship to the other application folders and, in particular, the relationship of their working folders to the working folders for other application folders.

Adding Existing Folders

You can add still more folders to a view by moving them or sharing them from other views. When a folder is moved or shared, it either keeps its absolute path or its relative path and is applied to its new parent folder. When a moved folder's path is relative, it usually ends up with a different working folder than it previously had. When a shared folder's path is relative, the shared folder has a different working folder in each location.

Note If a shared or moved folder's new working folder path exceeds the operating system's maximum working folder path length of 254 characters (including (\) backslashes), the application does not allow you to create the working folder.

For more information on moving and sharing folders, see "Moving Folders and Items Between Project Views" on page 193. This section explains only the relationships of folders that have been moved or shared with their new or additional parent folders and with their working folders.

Both the current view and the view from which the folders are moved or shared must use the same server configuration—and, therefore the same database and repository.

Locating Working Folders

Understanding the relationship between application folders and their working folders is important because the working folder stores the files that you check in and check out.

Each folder has a default working folder from which you modify working files. For team members that use the same folders, the working folder structure on one person's workstation is often the same as those on another person's workstation.

When you check out a file, the application copies the requested file revision to the appropriate working folder. If the working folder does not already exist on your workstation, the application automatically creates it for you as you check out files that go in that folder.

The application expects you to add and check in new file revisions from those working folders. If the working folder does not exist on your workstation, you can create it manually via Windows Explorer or automatically using the Create Working Folders command. After the working folder exists, you can add files to it.

The exact location of a working folder is displayed as one of the application folder's properties.

To determine the working folder for an application folder, select Properties from the Folder or context menu. The resulting *Folder Properties* dialog displays the path in the *Complete Working Folder Path* display box.

To create a particular working folder on your workstation:

1 Select the appropriate folder from the folder hierarchy.

2 Do one of the following:

- Check out a file from the folder.

- Select Create Working Folders from the *Folder* or context menu.

Once the working folder exists, you can copy files to it or create files in it and add them to the application.

Note If you have shared or moved a working folder path and the new path length exceeds the operating system's maximum working folder path length of 254 characters (including (\) backslashes), the application does not allow you to create the working folder.

Using an Alternate Working Folder for the View

This section explains how to select alternate working folders at the view level. Your alternate selection can affect all the folders on the workstation whose working folders' paths are relative to the application's root folder.

In many cases, team members prefer to have the same working folder structure on every workstation. However, this is not always possible. For example, one workstation may have a drive C:\ while another may have a D:\.

Note You can also create alternate working folders for individual application folders. This change affects the selected folder and many, if not all, of its child folders. See "Using an Alternate Working Folder" on page 47 for details.

To create an alternate working folder for a view:

1. Open the view.
2. Select View > Properties from the menu bar. The *View Properties* dialog appears.
3. From the *Working Folder* group box, select the Alternate option button.
4. Enter the absolute path to a working folder or click Browse to locate a working folder.
5. Click OK.

Adding Folders to a View

If you add a folder to a view, its working folder can be any of the following:

- Any folder on your workstation specified by you.
- A non-existing working folder specified by you and created by the application on your workstation. If the existing folder has child folders, one or more of them can also be added to the view.
- A child of the parent application folder's working folder. If you do not specify a working folder, the application appends the new folder's name to its parent's complete working folder path.

Note If the parent folder's working folder path length exceeds the operating system's maximum working folder path length of 254 characters (including (\) backslashes), the application does not allow you to create the new working folder.

The newly added folder assumes the parent folder's behavior, with a few exceptions. For example, the child folder might have the Branch On Change check box disabled because it makes no sense for this folder to branch.

You cannot add a folder to a view if the parent folder is read-only. Reasons why the parent folder might be read-only include:

- The parent folder is part of a read-only reference view.

- The parent folder is part of a view that is currently configured to a specific label, promotion state, or date and time (all, therefore, in the past and unchangeable).
- The parent folder itself is configured to a specific label, promotion state, or date and time (all, therefore, in the past and unchangeable).
- The parent folder's behavior has the Branch On Change check box enabled but cleared and the corresponding folder in the parent view is read-only. In a new branching view, the folders, files, etc. do not branch until the Branch On Change box is selected. Until then, changes made to the folder are really being made to the corresponding folder in the parent view. If the corresponding folder in the parent view is read-only, no changes can be made.

To add a new folder to a view:

1. Select New from the Folder or context menu. The *New Folder Wizard* dialog appears.
2. Select a folder from the folder tree and click Next. The new folder will be created as a child of the selected folder.

   The *New Folder Wizard: Folder Name* dialog appears.
3. Enter a name for the child folder in the *Folder Name* text box. Use a maximum of 254 characters.
4. Do one of the following:
   - Leave the *Working Folder* text box blank. The application creates the working folder using the name of the new folder and the path to its parent's working folder.
   - Enter or browse for the path to an existing working folder in the *Working Folder* text box.

     When you browse for a path, you create an absolute path to this folder's working folder.

     Important If you want this folder's working folder to be relative to its parent folder's working folder, enter the addition to that path in this text box. Do not browse.
5. In the Folder Description text box, enter a description of up to 254 characters for the new folder.
6. Click Next to continue.

   The *New Folder Wizard: Child Folders* dialog displays the new folder in the *New Folder's Child Folders* box.

   If the working folder has child folders, an application folder is created for each of them.
7. Do one of the following:
   - To exclude a child folder from your project, select the folder and click Exclude.
   - To exclude all child folders, click Exclude All. To redisplay folders you have excluded, click Reset Folders.
8. Click Next. The *New Folder Wizard: Folders* dialog displays the view's folder hierarchy with the new child folder.
9. Click Finish.

Tip Another way to add a folder to a project view is by dragging a folder from Explorer or Network Neighborhood to an existing folder in a project view. Also, selecting a folder in the project view before dragging the folder from Explorer or Network Neighborhood ensures that the selected folder becomes the parent of your new folder. The new folder's working folder is the location to which you dragged the folder.

Changing a Folder Name or Description

Use the *Folder Properties* dialog to change folder names and descriptions.

To change a folder's name or description:

1. Select the folder from the folder hierarchy.
2. Select Properties from the *Folder* or context menu. The *Folder Properties* dialog appears.
3. Select the Name tab.
4. Change the name or description or both.
5. Click OK.

Excluding Files from Folders

Use the *Folder Properties* dialog to exclude certain files or types of files from visibility. If a working file in an application folder's working folder would have the status Not In View but it matches a file specification in one of the exclude lists, the application does not display it at all. It is as though the file did not exist.

For example, suppose you are creating files in an application that makes automatic backup copies of each file (with the extension .bak) every time you save a file. Your working folder might contain several .bak files, but you have no reason to add those files to the project view. From the application, it is annoying to see these .bak files as possible candidates, so you exclude them. Excluding files is done on a per-folder basis. However, exclude lists can be inherited from parent folders. \

To exclude files from a folder:

1. Select the folder from the folder hierarchy.
2. Select Properties from the *Folder* or context menu.

   The *Folder Properties* dialog appears.
3. Select the Exclude tab.
4. Do one of the following:
   - Select the Inherit And Use Local Exclude Lists option button to exclude files that match this folder's exclude list as well as the exclude lists available to its parent folders.
   - Select the Local Exclude List option button to exclude files only if they match this folder's exclude list.
   - Select the No Exclude List option button to make sure that all files are included
5. Click OK.

Note The Exclude tab has no effect on files that are already part of the project—only those with Not In View as their status.

Creating Folders for Other Operating System Files

The application makes it easy to apply version control to files managed by other operating systems. You simply map a location on a computer running UNIX or another operating system as a drive on a computer running Windows NT, 2000, or XP. The mapping can be made using any of the following:

- The Microsoft's Services for UNIX package

- Another third-party NFS (Sun Network File System) services package that permits Windows systems to connect with computers sharing files via NFS services
- Samba on a UNIX or other operating system to provide Windows SMB (System Message Block) file sharing services from that system

After the drive is mapped, you can create an application folder for that drive and put the files at that location under version control.

However, this affects both file names and the end-of-line (EOL) marker used in text files because:

- Windows platforms use the carriage return/line feed (CR/LF) combination as the EOL character while UNIX uses a line feed (LF) and some other operating systems use just a carriage return (CR).
- UNIX file names (and those of some other operating systems) are case sensitive, while Windows file names are *not* case sensitive. For example, foo.txt and Foo.txt are different files on the UNIX and other platforms, but the same file in Windows.

So, in addition to creating the folder, you must set the folder properties to indicate the appropriate EOL character and case sensitivity. The EOL character and case sensitivity settings that you select become the settings for the selected folder and all of its child folders. However, you can change these settings on a folder-by-folder basis if you need to.

Note You can also check UNIX files in and out using the command line (after installing the command line that comes with the Cross-Platform client). For details, see "Using the stcmd Command-line Interface" on page 285.

The Cross-Platform client supports EOL conversions using personal options. For more information, see "Setting File Options" on page 275.

For information about using folder properties in view profiles, see "Using View Profiles" on page 25.

To create a folder (or branch of folders) for use with UNIX or another operating system's files:

1. With the aid of NFS or Samba services, connect a computer running UNIX or another operating system as a drive on your workstation.
2. Create a folder for that drive (or one of its child folders) using the procedure in "Adding Folders to a View" on page 44.
3. Select the folder from the folder hierarchy.
4. Select Properties from the *Folder* or context menu. The *Folder Properties* dialog appears.
5. Select the Files tab.
6. In the EOL Conversion group box, select the Override For This Folder And All Its Child Folders check box.
7. Select the option button for the appropriate EOL character.
8. In the *File Path Conventions* group box, select the Override For This Folder And All Its Child Folders check box.
9. Select the Case Sensitive File Paths check box.
10. Click OK.

Using an Alternate Working Folder

Your administrator sets the initial working folder for the view and, therefore, for most of the child folders in that view. As team members add application folders, they also

select working folders for those folders. However, you can use alternate working folders on your workstation. For example, suppose you are the technical writer and the working folder for the application folder named User Manuals is C:\orion\documentation\manuals. If you would rather use a shorter path or a path that you have previously used for manuals, such as C:\manuals, you set an alternate working folder for yourself in the application.

Note If you have already been using a working folder for a particular folder, see also "Changing a Working Folder" on page 48.

To use an alternate working folder:

1. Select the folder from the folder hierarchy.
2. Select Properties from the *Folder* or context menu.The *Folder Properties* dialog appears.
3. Select the Name tab.
4. In the *Working Folder* group box, select the Alternate option button.
5. Specify a more convenient location on your workstation's hard drive for the working folders for an application folder and its child folders. For example, you might want to use a shorter path or a different drive letter.
6. Click OK.

Note The alternate working folder path is specific to the workstation. For example, if you log onto the project as another user on this workstation, the application uses the alternate folder. If you log on as yourself on another workstation, your alternate working folder setting is not known.

Changing a Working Folder

After you have been using a working folder, changing to another working folder for that same application folder can cause your file status information to be lost—unless you perform the following procedure. You change the working folder by:

- Changing the default working folder.
- Using an alternate working folder. See "Using an Alternate Working Folder" on page 47.
- Changing the alternate working folder.

It is best to check in modified and new files before you make this change. Then you check the files out to the new location without any loss of status information.

If your administrator changes the working folder, check out files from the new working folder. Then copy any modified or new working files to the new location using Windows Explorer, for example.

Caution If you move one or more working folders from one location on your hard drive to another and, afterwards, change the working folder in the application, the status of most of the files in the working folder become Unknown and cannot be corrected using File > Update Status. However, you can do a force check out to get tip revisions or a force check in to make your working files the tip revisions.

To change the default or alternate working folder:

1. Do one of the following:
   - If you are changing the working folder, check in the files that need to be checked in (for example, those with the statuses Modified, Merge, or Not In View) before you make the change.
   - If the change has already been made, for example, by your administrator, make a note of the files that need to be checked in.

2 Select the folder from the folder hierarchy.

3 Select Properties from the Folder or context menu. The *Folder Properties* dialog appears.

4 Select the Name tab.

5 Do one of the following:

- Select the Default option button to change the default working folder path.
- Select the Alternate option button to change the alternate working folder path.

6 Do one of the following:

- Enter a relative path.
- Enter or browse for an absolute path.

7 Click OK.

8 Check out the files to the new working folder.

9 (If necessary) Copy the files that should have been checked in before this change to the new working folder and check them in.

Using Folder Properties in a View Profile

A view profile is a set of limitations applied to folders in a view. A profile specifies which folders are visible and which EOL and path case sensitivity settings apply to those folders. For example, if you are building a UNIX version and a Windows version of a product from the same view, you need a profile for each. UNIX expects paths to be case sensitive and uses a linefeed as the EOL (end-of-line) character in a text file, while Windows paths are *not* case sensitive and Windows text files use the carriage return/linefeed combination to indicate the end of a line.

For more information, see “Using View Profiles” on page 25.

Chapter

6

Managing Data in the Upper Pane

Displaying the most useful data in the upper pane is a combination of:

- Selecting the right fields (or properties) as column headers
- Sorting and grouping the rows of items appropriately
- Applying a meaningful query so that you only see rows of interest
- Creating filters to make the above three bulleted items repeatable

Filter Performance

We recommend displaying the same fields in most filters for a component. When users switch from one filter to another, if the fields displayed in the new filter are different, the entire client cache reloads.

If you have lots of items in the component and lots of extra memory on the server, you should consider turning on server-side caching for the component. Server-side caching is an advanced setting that can only be set in starteam-server-configs.xml. See the *StarTeam Administrator's Guide* for details about the FilesCaching (on by default), ChangeRequestCaching, Tasks Caching, and TopicsCaching options.

Controlling the Columns

You can change what column headers appear in the upper pane. This affects what you see in the upper pane and what you can find using the Find command.

To change column headers, do one of the following:

- Right-click a column header on the upper pane, then select Show Fields from the context menu.
- Select Filters > Show Fields from an appropriate item menu or context menu. For example, if files are displayed in the upper pane, you can use the File menu.

The *Show Fields* dialog displays two lists. The *Available Fields* list box contains all the fields that could be displayed as column headers—but that are not currently displayed. The *Show These Fields In This Order* list box displays all the fields that are currently displayed.

For a list of the fields available for each item, see “Understanding the Fields” on page 65.

Note It is possible to see more than one field with the same name in the Available Fields list. If you have Enterprise or Enterprise Advantage, most fields’ display names can be customized and the Server does not force them to be unique. Because this can cause confusion, duplicate display names should be avoided.

To display additional fields in the upper pane.

1 Select the fields to appear as the column headers from the *Available Fields* list.

2 Click Add.

To stop displaying fields in the upper pane.

1 Select the fields to be removed from the *Show These Fields In This Order* list box.

2 Click Remove.

To change the order of the fields to be displayed in the upper pane.

Drag each field name to the desired location in the Show These Fields In This Order list box.

Tip Double-clicking a field name moves it from one list box to the other.

The *Show Fields* dialog initially displays the most commonly used fields. Select the Show Advanced Fields check box to select from a complete list of the available fields.

Sorting and Grouping the Data

Clicking a column header sorts the data in the upper pane based on the value in that column. The sort is in ascending order by number, letter, internal order, or internal key, depending on the data. You can also sort the data in the lower pane when the Link tab is selected.

To reverse the sort order from ascending to descending, click the header a second time. A triangle appears on the sorted column’s column header. It points upward for ascending sorts and downward for descending sorts.

Use Expand All or Collapse All on the File, Change Request, Topic, Task, or Audit menu to expand or collapse all the groups.

To perform more than a primary sort (based on one column) and/or group items with the same value in a column, use the following procedure.

To perform up to a fourth-order sort:

1 Do one of the following:

- Right-click a column header on upper pane, then select Sort and Group from the context menu.
- Select Filters > Sort and Group from the appropriate menu or context menu.

The *Sort and Group* dialog displays four group boxes, each indented slightly more to the right than the one above it. The first group box designates a primary sort order, the second designates a secondary sort, and so on.

2 (Optional) To list all the fields in First By and Then By drop-down list boxes, select the Show Advanced Fields check box at the bottom of the dialog. Some fields are rarely used and considered advanced.

3 Select a field from the *First By* drop-down list box.

For a list of the fields available for each item, see “Understanding the Fields” on page 65. If you are grouping the items, the field does not need to be displayed in the upper pane. If you are not grouping the items, you can sort them based on a field

that is not displayed, but you will not be able to tell where one group leaves off and the next begins.

4 Select the Ascending or Descending option button. The default setting is ascending order.

5 Select Group By to group the items which have the same values in this field.

Important If you do not select any additional sort options, text fields are sorted in ASCII order. Enumerated and user ID fields are sorted by their internal order or internal keys. That is, enumerated fields are sorted in the order given to them by the person who created the field; user ID fields are sorted in the order in which they were created. The application disables the Sort Options button for numeric and date/time fields.

6 (Optional) Select Sort Options for additional sorting selections. The *Sort Options* dialog appears.

- Select As Text to sort enumerated and user ID fields by the names of their possible values. For text fields, As Text is your only choice.
- Clear the Case-sensitive check box to sort alphabetically or select it to sort in ASCII order (where uppercase letters precede lowercase letters).

7 Repeat steps 3 through 5 to add secondary and lower order sorts by using the Then By group boxes.

Using Queries

You can use a query to limit the items displayed in the upper pane to those that match the query. Each query is performed on all of the items in the part of the folder hierarchy you have selected. The fields included in your query do not have to be displayed in the upper pane.

Queries have the following characteristics:

- A name that easily identifies the query
- Public or private status

  Public queries can be used by anyone with the access rights to use them. Private queries are only displayed on your workstation.
- A logical expression to be applied to items of a particular type, such as files or change requests

Queries appear in every project view within the same server configuration. For example, if Project A and Project B are in the same server configuration, you can use the same queries in each project.

In the *Queries* dialog, queries are listed in the order in which they were created. This list is *not ca*se sensitive. For example, if you have a query named “recent CRs”, you cannot create a query named “Recent CRs” or “recent crs”. The application considers them to be the same query. You may have both a public and private query with the same name; they are differentiated by the icon.

Note After a query has been saved in either a public or private state, its state cannot be changed. However, you can copy a query and change the state of the new query. Because you can use only private queries in private filters and only public queries in public filters, if you copy a filter and change the state of the new filter, the application displays an error message—unless no query is associated with the filter.

Applying an Existing Query

You or your administrator may create queries that you will use frequently. For example, you might create a query that locates all change requests that you are responsible for. After a query has been created, you can use it to:

- List in the upper pane only those items that match the query.
- Select items from the upper pane that match the query—even though all the other items are still displayed.

  Usually you select items because you want to use them as a group. For example, you can print a report about them, flag them, attach a label to them, or check them out (if they are files).

The status bar indicates whether a query (other than the one that is part of the filter) has been applied to the upper pane.

To apply an existing query to the upper pane:

1. Do one of the following:
   - Right-click a column header on upper pane, then select Queries from the context menu.
   - Select Filters > Queries from the appropriate menu or context menu.

   The *Queries* dialog appears.
2. Select a query.
3. Click Select to apply the selected query to the contents of the upper pane.
4. If you do not like the effect of the query, reopen the *Queries* dialog and click Clear Query to return to the previously displayed list of items.

To select the items in the upper pane that match an existing query:

1. Choose Select > By Query from the appropriate menu or context menu.

   The *Select Query* dialog lists all existing queries.
2. Select the query of interest, and click OK. Items that match this query become selected in the upper pane.

   The status bar indicates the number of items in the upper pane and the number of items selected from the pane (for example, 8 files, 1 selected).

Creating a Query

Before you create a query, you need to be familiar with writing queries and with using the tree in the *Query* dialog.

A query is a logical expression that consists of:

- One or more conditions

  A condition consists of a field (not necessarily one that is a current column header), a relational operator, and a value to be compared to the value of the field.

  For example, a condition used to locate change requests might be:

  ```
  Responsibility Equals Rhonda Thurman
  ```

  Responsibility is the name of the field, Equals is the relational operator, and Rhonda Thurman is the value to be located in the Responsibility field.
- The logical operators that bind the conditions together: AND, OR, and NOT.

For example, you can locate all the change requests for which Rhonda Thurman is responsible that also have a high severity. You need the following two conditions bound together with the logical operator AND:

```
Responsibility Equals Rhonda Thurman
Severity Equals High
```

Tip You can write simple queries that contain only one condition, or more complicated queries that use several conditions and one or more logical operators.

Creating a Simple Query

The simplest query has only one condition. For example, the following simple query locates change requests that someone named Rhonda Thurman is responsible for:

```
Responsibility Equals Rhonda Thurman
```

To create a simple query:

1. Select the appropriate item tab.
2. Do one of the following:
   - Right-click a column header on upper pane, then select Queries from the context menu.
   - Select Filters > Queries from the appropriate menu or context menu.

   The *Queries* dialog appears.
3. Click New. The *New Query* dialog appears.
4. Enter a name for your query in the *Name* text box.
5. Do one of the following:
   - Select the Public check box to add this query to the project so anyone can use it.
   - Clear the check box to make the query private, available only on your workstation.
6. Create the condition:
   a. Select a field from the *Field* drop-down list box.

      The *Field* list box lists the most common fields. To include every possible field in the list box, select the Show Advanced Fields check box.

      For a list of the fields available for each item, see "Understanding the Fields" on page 65.

   b. Select an operator from the *Operator* drop-down list. (For more information about the operators, see "Relational Operators" on page 65.)

   c. Enter or select a value in the Value text or list box. This value will be compared with the value of the selected field for each available item.
7. Click Add to place this condition in the *Query Tree* box.

   The default setting for the Query Tree box uses the AND operator. This operator is the root of the tree. For simple queries, where there is only one condition, such as the default, the application ignores the logical operator.
8. (Optional) Click View As Text to view the query in text format.
9. Click Save. The *Queries* dialog displays your query.
10. If this is a public query, you can set access rights for it. For more information, see the *StarTeam Administator's Guide*.
11. Do one of the following:

- Select the query from the *Queries* list and click Select to apply the query to the items in the upper pane.
- Click Close to exit without applying your query.

You can edit and copy existing queries using the *Queries* dialog. For example, if you change your mind about the condition while you are creating your query, select the condition in the *Query* dialog, and then click Edit to modify it or Delete to eliminate it.

Select Copy to create another query using conditions from an existing query. Then you can edit and add conditions to the new query.

Be careful not to AND mutually exclusive conditions. For example, in the following query, both Jack Winter and Lin Hu could not both have entered the same change requests, so nothing matches the query.

```
Created By Equals Jack Winter AND Created By Equals Lin Hu.
```

Creating a More Complex Query

The application allows you to construct queries with multiple logical operators.

In Algebra (or in the text format), you AND or OR pairs of conditions, but the application allows you to use any number of conditions per logical operator. For example, if your query is to find change requests with:

```
Responsibility Equals Rhonda Thurman

AND

Severity Equals High

AND

CR Number Same or Greater 3000
```

The text format would display:

The *View as Text* dialog displays two logical operators, while the query tree would only need to display one as follows:

The NOT operator can precede any condition or combination of conditions joined by ANDs and ORs.

The *View as Text* dialog will display three logical operators, the same number as in the text but arranged slightly differently.

Important Nodes farthest from the root are evaluated first and the root node is evaluated last.

To create a more complex query:

1. Follow steps 1 through 4 in "Creating a Simple Query" on page 57. By default, the first logical operator is AND.
2. (Optional) If the first logical operator in your query should be OR, select the AND logical operator in the query tree; then click the AND->OR->NOT button. This changes an AND to an OR. Similarly, one more click changes the OR to a NOT. You click the button until the operation becomes the one you want to use.
3. Create a condition for this logical operator:
   a. Select a field from the *Field* drop-down list box.

      The *Field* list box lists the most common fields. To include every possible field in the list box, select the Show Advanced Fields check box.

      For a list of the fields available for each item, see "Understanding the Fields" on page 65.

   b. Select an operator from the *Operator* drop-down list. (For more information about the operators, see "Relational Operators" on page 65.)

   c. Enter a value or select one from the Value text or list box. This value will be compared with the value of the selected field for each available item.
4. Click Add to place this condition in the *Query Tree* box.
5. To add more conditions for this operator, repeat steps 3 and 4 for each additional condition. Then go on to step 6.
6. (Optional) To add another logical operator:
   a. Select an existing logical operator (the one that should precede the new operator in the query and, therefore, be its parent in the tree).

   b. Click the AND**,** OR, or NOT button.

   c. If the operator is AND or OR, repeat steps 3 through 5.

      If the operator is NOT, repeat step 6.
7. (Optional) Repeat step 6 for additional logical operators.

   You can rearrange nodes in the query tree using drag-and-drop.

Tip

- It is best to use the condition or logical operation that will result in the fewest matches as the first condition or logical operation.

Troubleshooting a Query

To avoid creating an incorrect query, it is important to plan ahead and draw out a tree for you query.

Although normally you can drag a logical operator to a new location, you cannot drag a logical operator to the root of the query tree. To repair the earlier query, you change the AND to OR and the OR to AND and drag the conditions to the appropriate locations.

Editing a Query

You might have to edit or add to a query to display a more useful set of data.

To edit a query:

1 Do one of the following:

- Right-click a column header on upper pane, then select Queries from the context menu.
- Select Filters > Queries from the appropriate menu or context menu.

The *Queries* dialog appears.

2 Select the query to be edited from the list, then click Edit. The *Edit Query* dialog appears.

3 Edit the appropriate nodes of the tree, then click Save.

Copying a Query

The copy query feature enables you to creating new queries based on existing queries. Instead of recreating existing conditions, you can copy them.

To copy a query:

1 Do one of the following:

- Right-click a column header on upper pane, then select Queries from the context menu.
- Select Filters > Queries from the appropriate menu or context menu.

The *Queries* dialog appears.

2 Select the query to be copied from the list then click Copy.

The *Copy Query* dialog appears.

3 Enter the name for the new query in the Query Name text box.

4 Do one of the following:

- Select the Public check box to add this query to the project so anyone can use it.
- Clear the check box to make the query private, available only on your workstation.

5 Click OK. This action displays the query in the Query list.

6 If this is a public query, you may be able to set access rights for it. For more information, see the *StarTeam Administator's Guide*.

7 Select the query from the list and click Edit to change the conditions in the query.

Deleting a Query

You can delete queries that you no longer use.

To delete an existing query:

1 Do one of the following:

- Right-click a column header on upper pane, then select Queries from the context menu.
- Select Filters > Queries from the appropriate menu or context menu.

The *Queries* dialog appears.

2 Select the query to be deleted.

3 Click Delete.

A message box asks you to confirm your deletion.

4 Click Close.

Using Filters

A filter is a named arrangement of data. It consists of a set of fields (for column headers), sorting and grouping information, and a query.

Filters are listed in alphanumeric order. This list is *not* case sensitive. Thus, if you have a filter named “recent CRs”, you cannot create a filter named “Recent CRs” or “recent crs”.

To filter the data in the upper pane, you can:

- Apply an existing filter.
- Arrange the data yourself by changing the displayed fields, sorting and grouping the files, and applying a query (as explained in the preceding section).

  After arranging the data, you can create a filter based on that arrangement. Doing this makes it easier to create the same arrangement another time.
- Create a new filter from scratch.

After a filter has been created, it appears in every project view within the same server configuration. For example, if Project A and Project B are in the same server configuration, you can use the same filter in each project’s view.

You can change the order in which filters appear in the Filters list box by naming or renaming them. For example, when the Files tab is selected, the filters for files appear in the Filters drop-down list box in alphanumeric order. Suppose the filter named <All Files By Status> is the first in the Filters list box for files. If you want another filter to be the first in the list box, give it a name (or save it with a new name) that places it at the top of the list.

Note After a filter has been saved in either a public or private state, its state cannot be changed. Because you can use only private queries in private filters and only public queries in public filters, you cannot copy of a filter and change the state of the new filter unless no query is associated with the filter.

Applying an Existing Filter

Existing filters can be private (only for your use) or public (to be used by any team members with the appropriate access rights).

The application allows you to apply an existing filter in two ways:

- By selecting it from the Filters drop-down list box on the toolbar
- By selecting it from the *Filters* dialog

To apply a filter from the *Filters* dialog:

1. Do one of the following:
   - Right-click a column header on upper pane, then select Filters from the context menu.
   - Select Filters > Filters from the appropriate menu or context menu.

   The *Filters* dialog appears.
2. Select a filter from the dialog.
3. Click Select to apply the filter to the data.

Creating a Filter

You can create filters to make your job easier. For example, if you are a QA supervisor, you may need to check how many change requests have been found in a particular time frame.

To save the current arrangement of data as a filter:

1. Do one of the following:
   - Right-click a column header on the upper pane, then select Save Current Settings from the context menu.
   - Select Filters > Save Current Settings from the appropriate menu or context menu.

   The *Save Current Settings* dialog appears.
2. In the Filter Name text box, enter a name for this filter.
3. Do one of the following:
   - Select the Public check box to add this filter to the project so anyone with the appropriate access rights can use it.
   - Clear the check box to make the filter private, available only for your use.

To create a new filter from scratch:

1. Do one of the following:
   - Right-click a column header on upper pane, then select Filters from the context menu.
   - Select Filters > Filters from the appropriate menu or context menu.

   The *Filters* dialog appears.
2. Click New. The *New Filter* dialog appears.
3. In the Filter Name text box, enter a name for this filter.
4. Do one of the following:
   - Select the Public check box to add this filter to the project so anyone with appropriate access rights can use it.

- Clear the check box to make the filter private, available only for your use.

5 Click OK to return to the *Filters* dialog.

6 Select any or all of the following options:

- Fields to select the column header fields. To do this, follow the steps in "Controlling the Columns" on page 53.
- Sort, Group to sort and group items up to four categories in ascending or descending order. To do this, follow the steps in "Sorting and Grouping the Data" on page 54.
- Query to limit the items that appear in the upper pane to those that match the query. Follow the steps in "Creating a Query" on page 56.

7 If you are creating a filter for files:

a Click Context to limit the files affected by the filter to those in the view, not in the view, or to all non-excluded files.

The *Set Filter Type* dialog appears.

b Apply the filter to one of the following by selecting an option button:

- Items In The View

  This is equivalent to applying both your filter and the Files In View filter.
- Items Not In The View

  This is equivalent to applying both your filter and the Files Not In View filter.
- All Items Not Excluded From The View

  This is equivalent to applying both your filter and the <All Non-Excluded Files> filter.

8 Click OK to return to the *Filters* dialog.

9 If this is a public filter, you may be able to set access rights for it. For more information, see the *StarTeam Administator's Guide*.

10 Do one of the following:

- Click Select to apply the filter to the upper pane.
- Click Close to exit without applying your filter.

Tip You can create a new filter based on an existing filter by selecting a filter from the *Filter* dialog, clicking Save As to rename it, then editing the filter's parts as appropriate.

Editing a Filter

You edit filters by changing their fields, sort orders, or queries.

To edit a filter:

1 Do one of the following:

- Right-click a column header on upper pane, then select Filters from the context menu.
- Select Filters > Filters from the appropriate menu or context menu.

The *Filters* dialog appears.

2 Select a filter from the list box.

3 Follow steps 4 through 9 in "Creating a Filter" on page 62.

4 Click Save As. The *Save Filter As* dialog appears. Do not change the filter name.

5 Click OK to return to the *Filters* dialog.

6 Do one of the following:

- Click Select to apply the edited filter to the upper pane.
- Click Close to exit without applying the edited filter.

Copying a Filter

You can copy a filter using the Save As button in the *Filters* dialog. If you copy a filter that is associated with a query, the new filter must have the same state (public or private) as the original filter. This is because only public queries can be used with public filters and only private queries can be used with private filters.

To copy a filter:

1 Do one of the following:

- Right-click a column header on upper pane, then select Filters from the context menu.
- Select Filters > Filters from the appropriate menu or context menu.

The *Filters* dialog appears.

2 Select a filter from the list box.

3 Click Save As. The *Save Filter As* dialog appears.

4 Give the filter a new name.

5 Select or clear the Public check box.

If a query is associated with this filter, the state of the new filter must be the same as the state of the original filter.

6 Click OK to return to the *Filters* dialog.

7 Do one of the following:

- Click Select to apply the edited filter to the upper pane.
- Click Close to exit without applying the edited filter.

To copy a filter that has an associated query and change its public/private state:

1 Copy the query first, giving the new query the appropriate state.

2 Detach the original query from the filter.

3 Make a copy of the filter.

4 Reattach the original query to the original filter.

5 Attach the new query to the new filter.

Resetting a Filter

You can apply a filter, rearrange the data, and return to the current filter as it is defined on the server.

For example, suppose you select the <Show All> filter from the Filter drop-down list box. Next, you decide to perform additional functions to the data, such as sorting the data in a new way, showing an additional column of data, or applying a new query. An asterisk appears in front of the filter's name (*<Show All>) indicating that your original filter choice has been altered. After you find out what you wanted to know about the rearranged data, you can restore the current filter to its original definition using the Reset Current Filter command.

To reset a filter, do one of the following:

- Right-click a column header on the upper pane, then select Reset Current Settings from the context menu.
- Select Filters > Reset Current Settings from the appropriate menu or context menu.

Deleting a Filter

You can delete filters that you no longer use.

To delete a filter:

1. Do one of the following:
   - Right-click a column header on upper pane, then select Filters from the context menu.
   - Select Filters > Filters from the appropriate menu or context menu.

   The *Filters* dialog appears.
2. Select the filter to be deleted from the Filters list.
3. Click Delete.

   A message box asks you to confirm your deletion.
4. Click OK to return to the *Filters* dialog.
5. Click Close.

Setting Access Rights for Queries and Filters

Access rights can be set for public queries and filters. For more information, see the *StarTeam Administator's Guide*.

Understanding the Fields

The following sections describe both the common and advanced fields that you can display as columns in the upper pane or use in queries, and the relational operators that you can use with those fields to define conditions. To see an advanced field in the Show Fields, Sort and Group, or Query dialog boxes, you must select the Show Advanced Fields check box.

The field descriptions presented in the following sections includes the internal identifier for this field. In report templates, you must use the internal identifier, instead of the field name. Most internal identifiers contains no spaces. However, some do have spaces. Internal identifiers are case sensitive.

Relational Operators

The relational operators that you can use to define conditions in a query vary according to the type of field.

The text operators used on text fields are:

- Equals
- Is Not
- Contains (ignore case)
- Contains (match case)

- Starts with (ignore case)
- Starts with (match case)
- Ends with (ignore case)
- Ends with (match case)

The relational operators used on Boolean, enumerated type, and numeric fields are:

- Less Than
- Same or Less
- Equals
- Same or Greater
- Greater Than
- Is Not

The date/time operators used on date/time fields are:

- Before
- On or Before
- On
- On or After
- After
- Not On
- Before Date
- On or Before Date
- On Date
- On or After Date
- After Date
- Last (?) Days
- Last (?) Weeks
- Older Than (?) Days
- Older Than (?) Weeks

The Before, On or Before, On, On or After, After, and Not On operators compare both the date and the time parts of date/time fields. The Before Date, On or Before Date, On Date, On or After Date, and After Date operators compare only the date part of date/time fields.

The Last (?) Days and Last (?) Weeks operators match all dates starting with the date that was the specified number of days or weeks ago. The Older Than (?) Days/Weeks operators match all the dates prior to and including the date that was the specified number of days or weeks ago.

Note In date fields, the application treats blanks as zeroes. That means that “no date” is less than any specific date. For example, if you write a query that searches for change requests that were closed prior to some specific date, all the change requests with no date in the Closed On field are included in the results—even though they have not been closed yet.

It is easy to eliminate the change requests that contain blanks in the Closed On field from such a query. You simply AND the condition that searches for change requests closed on or before a specific date with another condition that searches for change requests closed after the date zero.

The following sections list the fields that are associated with particular types of items. You can use these fields to:

- Display any number of these fields as column headers in the upper pane. See “Controlling the Columns” on page 53 for more information.
- Sort and group the rows in the upper pane. See “Sorting and Grouping the Data” on page 54 for more information.
- Create queries. See “Using Queries” on page 55 for more information.
- Modify the default reports to suit your needs. See “” on page 253 for more information.

Remember that fields for different types of items (for example, a field and a change request) may have the same names, but their uses may be different.

File Fields

This section lists all the file fields in alphabetical order.

- Archive Format (Advanced)

  Values: Native-I, Native-II

  Internal Identifier: ArchiveFormat

  Indicates whether a file’s revisions are stored in a Native-I or Native-II format.

- Archive Path

  Values: text

  Internal Identifier: ArchivePath

  The path to the PVCS archive or VSS project containing a file.

- Archive Name (Advanced)

  Values: 32-digit hex string representing the MD5 value of file

  Internal Identifier: StArchiveName

  For a file stored in a Native-II Vault, indicates the name of the file that stores the tip revision. This name is the MD5 value of that file revision’s content, converted to a 32-digit hex string.

- Archive Type

  Values: Native, PVCS, VSS

  Internal Identifier: Type

  Indicates whether a file is stored as a StarTeam (Native), PVCS, or VSS file.

- Branch On Change (Advanced)

  Values: No, Yes

  Internal Identifier: BranchOnChange

  Indicates whether a file will branch when it changes.

  The value is No if the file’s behavior is not set to “Branch On Change.” Reasons for this may be:

  - The file is in the root or a reference view and the “Branch On Change” feature is disabled.
  - The file is in a branching view but has already branched as a result of a change, which, in turn, results in the “Branch On Change” feature becoming disabled.

- The file is in a branching view, but its behavior currently does not permit it to branch on change. This means that modifications are checked into the parent view.

Note If the value is No, the value of the Branch State explains the No.

The value Yes indicates that the file resides in a branching view and has its behavior set to “Branch On Change,” but has yet to be changed.

- Branch State (Advanced)

  Values: Branched, Not Branched, Root

  Internal Identifier: BranchState

  Indicates whether a file has branched in the child view, is still unbranched (and therefore is part of the parent view), or was created in the view in which it resides.

  The values Branched and Not Branched apply to files in branching views. The value Root applies to files created in the view in which the file currently resides.

  If the view is a reference view, it reflects the state of the file in the reference view’s parent.

- Comment

  Values: text

  Internal Identifier: Comment

  The initial 2000 characters provided as the reason for changing a file’s properties or contents are stored in the Short Comment field. This field, the Comment field, stores those 2000 characters and any additional text. Changing a file’s properties causes the application to create a new revision.

- CommentID (Advanced)

  Values: number

  Internal Identifier: CommentID

  The ID number assigned to revision comment. Displays -1 if no revision comment was supplied.

- Compression Level

  Values: Default, Maximize Compression, Maximize Speed, None

  Internal Identifier: Compression

  Indicates a file's level of compression.

  - Default: A compromise between Maximize Compression and Maximize Speed
  - Maximize Compression: The densest possible compression of file revisions to save space on the server
  - Maximize Speed: The fastest possible compression of file revisions to improve server performance
  - None: No compression

- Configuration Time

  Values: date/time

  Internal Identifier: ConfigurationTime

  Indicates the time to which a file is configured. If you configure a file to a specific time, this field contains that time. If you configure a file to a label or promotion state, this field shows either the time at which the label was created, or the time at which the label associated with the promotion state was created.

- Content Revision

  Values: number

  Internal Identifier: ContentVersion

  The number of times a file has been checked in. If the file is in a child view, it includes all the content revisions from the parent view in this number. Each revision appears in the file's history.

- Created By

  Values: list of users, <None>

  Internal Identifier: CreatedUserID

  The name of the user who created the first revision in the view. This is either the user who added the file to the project, or the user who checked in the revision that branched.

- Created Time

  Values: date/time

  Internal Identifier: CreatedTime

  The time at which the first revision in the view was created.

- Deleted By

  Values: list of users, <None>

  Internal Identifier: DeletedUserID

  The name of the user who deleted a file. Because deleted files do not appear in the list, this information is unavailable to users.

- Deleted Time

  Values: date/time

  Internal Identifier: DeletedTime

  The time at which a file was deleted. Because deleted files do not appear in the list, this information is unavailable to users.

- Description

  Values: text

  Internal Identifier: Description

  The description provided for a file at the time it was added to the view, including any later edits to it.

- Dot Notation

  Values: text

  Internal Identifier: DotNotation

  The branch revision number, for example, 1.2.1.0.

- Dot Notation ID (Advanced)

  Values: number

  Internal Identifier: DotNotationID

  The ID assigned to a particular branch revision number. For example, if a file was added to the current view (as opposed to inherited by the current view), its branch revision number is 1.x and its branch revision ID is 0. If a file was branched in the current view, its branch revision ID is dependent on the revision number in the parent view and the number of IDs already assigned in the current view. For example, if a file's revision number in the parent view is 1.7 at the time of the branch, and another file with that same parent revision number was given the Branch Revision ID 6, this file will also be given the Branch Revision ID 6.

- End Modified Time (Advanced)

  Values: date/time

  Internal Identifier: EndModifiedTime

  The date and time at which a revision ceased to be the tip revision. Although this field can be displayed in the upper pane, its value is always blank. This is because, at any given configuration time, the item is still the tip revision.

- EOL Character

  Values: numeric representation of ANSI character

  Internal Identifier: EOL

  For internal use only. This field is primarily used to determine an ANSI character to use when breaking up lines within files for delta storage with Native-1 Vaults.

- Executable

  Values: No, Yes

  Internal Identifier: Executable

  Indicates whether the executable bit should be set for a UNIX file.

- Extension

  Values: text

  Internal Identifier: Extension

  Displays the extension of the file.

- File Time Stamp at Check-In

  Values: date/time

  Internal Identifier: Modified

  The file's time stamp at the time it was last checked in.

- File Type

  Values: ASCII, Binary, Unicode

  Internal Identifier: Charset

  Indicates whether the file is an ASCII (text), binary, or Unicode.

- Flag

  Values: No, Yes

  Internal Identifier: Flag

  A flag specifically marks/bookmarks files in the upper pane on your workstation.

- Flag User List (Advanced)

  Values: byte array; displayed as a bracketed series of numbers in hex format. For example, [14 00 00 00] indicates a specific user.

  Internal Identifier: FlagUserList

  Cannot be used in queries. Identifies users who have set flags on a given item.

- Folder

  Values: text

  Internal Identifier: Folder

  The name of the folder with which a file is associated. This is not the name of the working folder.

- Folder Path

  Values: text

  Internal Identifier: Folder Path (contains spaces)

  The path to the folder with which a file is associated. This is not the path to the working folder.

- Hive ID (Advanced)

  Values: number assigned by the Server

  Internal Identifier: HiveID

  For a file stored in a Native-I Vault, indicates the ID number of the hive that stores the tip revision.

- Locked By

  Values: list of users, <None>

  Internal Identifier: ExclusiveLocker

  The name of user who has exclusively locked a file.

- MD5 Checksum

  Values: byte array; displayed as a bracketed series of numbers in hex format. For example, the Windows client may display [7B 05 71 BF 98 FA 40 0A D9 D7 6D 7A 25 89 12 67]. The Cross-Platform client displays only significant zeroes so the 05 and 0A would become just 5 and A. These clients display A–F as a–f.

  Internal Identifier: MD5

Cannot be used in queries. The MD5 checksum for the tip revision.

- Modified By

  Values: list of users, <None>

  Internal Identifier: ModifiedUserID

  The name of the user who last modified a file.

- Modified Time

  Values: date/time

  Internal Identifier: ModifiedTime

  The time at which a file was last modified. The file may have been checked in or had its properties changed. This has nothing to do with the working file. Use Local Time Stamp for the time a working file was last modified.

- My Lock

  Values: Exclusively Locked By Me, Non-exclusively Locked By Me, Not Locked By Me

  Internal Identifier: My Lock (contains space)

  Indicates whether the current user has the file locked and, if so, whether that lock is exclusive or not.

- Name

  Values: text

  Internal Identifier: Name

  Displays the name of the file.

- New Revision Comment (Advanced)

  Values: text

  Internal Identifier: New Revision Comment (contains spaces)

  Internal use only. the client uses this value during the item update process. The field always appears empty if added to the upper pane.

- Non-Exclusive Lockers

  Values: text

  Internal Identifier: NonExclusiveLockers

  The names of the users who have locked the file non-exclusively.

- Object ID

  Values: number

  Internal Identifier: ID

  Each file is assigned an object ID when it is added to a view. When it is branched in a child view, it is assigned another object ID. The original ID belongs to the file in the parent view.

- Parent Branch Revision (Advanced)

  Values: number

  Internal Identifier: ParentRevision

  The last digit in the branch revision number before a file branched. For example, if this number is 7, the branch revision was 1.7 at the time the file branched (becoming 1.7.1.0, as seen in the file's history). This number is -1 if a file was not inherited from the parent view.

- Parent ID (Advanced)

  Values: number

  Internal Identifier: ParentID

  The object ID of a file in the parent view. The Parent ID is -1 if this view has no parent view.

- Parent Revision (Advanced)

  Values: number

  Internal Identifier: PathRevision

  The revision number at which a file branched. For example, if this number is 8, this file's revision number in the parent view was 8 at the time the file branched. The history should show that revision 9 in the first revision in the current view. This number is 0 if this file was not inherited from the parent view.

- Path

  Values: text

  Internal Identifier: Path

  The path to a file's working folder.

- Project ID (Advanced)

  Values: number

  Internal Identifier: ProjectID

  The ID number assigned to a project. Within a server configuration, projects are assigned ID numbers in the order in which they are created. The first project has ID 0.

- PVCS Revision

  Values: text

  Internal Identifier: PVCSRev

  The file's revision number in PVCS's dot notation.

- Read Only (Advanced)

  Values: No, Yes

  Internal Identifier: ReadOnly

  Indicates whether the file's configuration is read-only (as in a rollback configuration of a view) and/or its behavior does not allow it to branch on modification. Do not confuse a read-only configuration (an application issue) with a read-only file (an operating system issue). A read-only file cannot be edited and saved to disk. A file whose configuration is read-only can be edited and saved to disk; it just can't be checked in.

- Revision

  Values: number

  Internal Identifier: ViewVersion

  The number of times a file has been checked in or had its properties changed. If the file is in a child view, it includes all the revisions from the parent view in this number.

- Revision Flags (Advanced)

  Values: 0

  Internal Identifier: RevisionFlags

  Internal use only.

- Revision on Disk

  Values: number

  Internal Identifier: SyncPathVersion

  The number of the revision that is currently in the working folder on your workstation. The application displays no number if the file's status is Missing.

- Root Object ID (Advanced)

  Values: number

  Internal Identifier: RootObjectID

  The object ID of the oldest ancestor of a file. For example, if a file was not inherited from a parent view, the root object ID is the same as its object ID. If it was inherited from a parent view, the root object ID is the Parent ID, or the parent's Parent ID.

- Share State

  Values: DerivedShare, Not Shared, Root Share

  Internal Identifier: ShareState

  Indicates whether this item is shared. Not Shared means that the item is not shared. Root Share means that the item is shared and this item is the original (or root) reference. DerivedShare means that the item is shared, but this item is not the original (or root) reference.

- Short Comment

  Values: text

  Internal Identifier: ShortComment

  Stores the initial 2000 characters provided as the reason for changing a file's properties or contents. Additional text is stored in the Comment field.

- Size

  Values: number

  Internal Identifier: FileSize

  The tip revision's size in bytes.

- Status

  Values: Current, Merge, Missing, Modified, Not In View, Out Of Date, Unknown

  Internal Identifier: Status

  Indicates the relationship between the copy of a file in your working folder and the tip revision in the repository.

- Storage Type (Advanced)

  Values: Delta, Full Versions

  Internal Identifier: StorageType

  For a file still stored in a Native-I vault, indicates whether the revisions are stored as forward deltas (Delta) or in entirety (Full Versions).

- Sync Branch Version

  Values: number

  Internal Identifier: SyncObjectVersion

  A field used to determine status. The last number of the branch revision that was most recently checked out to the working folder.

- Sync Content Version

Values: number

Internal Identifier: SyncContentVersion

A field used to determine status. The revision checked out as the working file or, if the file needs to be merged, a number higher than that.

- Sync Known

  Values: No, Yes

  Internal Identifier: SyncKnown

  A field used to determine status. Indicates whether the server knows the working file's relationship to the tip revision.

- Sync Local Size

  Values: number

  Internal Identifier: SyncSize

  A field used to determine status. The size of the working file in bytes.

- Sync Local Time Stamp

  Values: date/time

  Internal Identifier: SyncTime

  A field used to determine status. The time stamp for the working file.

- Sync MD5

  Values: byte array; displayed as a bracketed series of numbers in hex format. For example, [ED FA 08 46 78 15 A1 4E 90 35 1B 46 33 0B 06 9D] as displayed on the Windows client. The Cross-Platform client displays only significant zeroes, so 08, 0B, and 06 would become just 8, B, and 6. These clients display A–F as a–f.

  Internal Identifier: SyncMD5

  Cannot be used in queries. A field used to determine status. The MD5 checksum of the working file.

- Sync On Path To Root

  Values: No, Yes

  Internal Identifier: SyncOnPathToRoot

  A field used to determine status. When the working file is not based on the tip revision, this field indicates whether the server knows the relationship between the two. A Yes value in this field means that the working file needs to be merged or is out of date. A No value means that the relationship cannot be determined.

- Vault Branch Version (Advanced)

  Values: number

  Internal Identifier: VaultVersion

  The number of times a file has been checked in from the current view.

- Version (Advanced)

  Values: number

  Internal Identifier: RevisionNumber

  The last number in the branch revision number. For example, if the branch revision number is 1.3.1.2, the version is 2.

- View

  Values: list of views, <None>

Internal Identifier: ViewID

The name of the view in which the item last branched. For example, if a file is inherited from a parent view but is branched in a child view, the value of this field in the child view changes from the name of the parent view to the name of the child view for the revision that branched and subsequent revisions in the child view.

- Working File Exists

  Values: No, Yes

  Internal Identifier: LocalFileExists

  Indicates whether a copy of a file is in its working folder.

- Working File Size

  Values: number

  Internal Identifier: LocalSize

  The size of the working file.

- Working File Time Stamp

  Values: date/time

  Internal Identifier: LocalTimestamp

  The time stamp of the working file.

Change Request Fields

This section lists all the change request fields in alphabetical order.

- Addressed By

  Values: list of users, <None>

  Internal Identifier: AddressedBy

  Indicates the user who resolved a change request (resolved statuses are Cannot Reproduce, As Designed, Fixed, Documented, and Is Duplicate).

- Addressed In

  Values: list of view labels, <None>

  Internal Identifier: AddressedIn

  Indicates the next build label created and applied to the view after the resolution to a change request occurs.

- Addressed In View

  Values: list of views, <None>

  Internal Identifier: AddressedInView

  Indicates in what view the change request has been resolved. This is important for shared, and perhaps moved, change requests.

- Attachment Count

  Values: number

  Internal Identifier: AttachmentCount

  The number of files attached to a change request.

- Attachment IDs (Advanced)

  Values: byte array; displayed as a bracketed series of numbers in hex format. For example, [00 00 00 00 02 00 00 00] indicates two specific attachments.

Internal Identifier: AttachmentIDs

Cannot be used in queries. The ID numbers assigned to attachments. For example, the first attachment within a project is 00 00 00 00.

- Attachment Names

  Values: text containing a series of file names separated by spaces

  Internal Identifier: AttachmentNames

  The names of the files attached to a change request.

- Branch On Change (Advanced)

  Values: No, Yes

  Internal Identifier: BranchOnChange

  Indicates whether a change request will branch when it changes.

  The value is No if the change request's behavior is not set to "Branch On Change." Reasons for this may be:

  - The change request is in the root or a reference view and the "Branch On Change" feature is disabled.
  - The change request is in a branching view but has already branched as a result of a change, which, in turn, results in the "Branch On Change" feature becoming disabled.
  - The change request is in a branching view, but its behavior currently does not permit it to branch on change. This means that modifications are checked into the parent view.

Note If the value is No, the value of the Branch State explains the No.

  The value Yes indicates that the change request resides in a branching view, has its behavior set to "Branch On Change," but has yet to be changed.

- Branch State (Advanced)

  Values: Branched, Not Branched, Root

  Internal Identifier: BranchState

  Indicates whether a change request has branched in the child view, is still unbranched and, therefore, a part of the parent view, or was created in the view in which it resides.

  The values Branched and Not Branched apply to change requests in branching views. The value Root applies to files created in the view in which the change request currently resides.

  If the view is a reference view, it reflects the state of the change request in the reference view's parent.

- Category

  Values: text

  Internal Identifier: Category

  Text identifying the subcomponent in which the defect occurs. It is usually used in combination with the Component field.

- Closed On

  Values: date/time

  Internal Identifier: ClosedOn

  The date and time at which a change request was closed.

- Comment

  Values: text

  Internal Identifier: Comment

  The initial 2000 characters provided as the reason for changing a change request's properties are stored in the Short Comment field. The Comment field stores those 2000 characters and any additional text. Changing a change request's properties causes the application to create a new revision.

- CommentID (Advanced)

  Values: number

  Internal Identifier: CommentID

  The ID number assigned to revision comment. Displays -1 if no revision comment was supplied.

- Component

  Values: text

  Internal Identifier: Component

  Text identifying the component in which the defect occurs. It is often used with the Category field to narrow that identification to a subcomponent.

- Configuration Time

  Values: date/time

  Internal Identifier: ConfigurationTime

  Indicates the time to which a change request is configured. If you configure a change request to a specific time, this field contains that time. If you configure a change request to a label or promotion state, this field shows either the time at which the label was created or the time at which the label associated with the promotion state was created.

- CR Number

  Values: number

  Internal Identifier: ChangeNumber

  The number assigned to a change request. For example, if the Object ID is 0, the change request number is 1.

- Created By (Advanced)

  Values: list of users, <None>

  Internal Identifier: CreatedUserID

  The name of the user who created the first revision in the view. This is either the user who initiated the change request or the user who modified the revision that branched.

- Created Time (Advanced)

  Values: date/time

  Internal Identifier: CreatedTime

  The time at which the first revision in the view was created.

- Deleted By

  Values: list of users, <None>

  Internal Identifier: DeletedUserID

The name of the user who deleted a change request. Because deleted change requests do not appear in the list, this information is unavailable to users.

- Deleted Time

  Values: date/time

  Internal Identifier: DeletedTime

  The time at which a change request was deleted. Because deleted change requests do not appear in the list, this information is unavailable to users.

- Description

  Values: text

  Internal Identifier: Description

  The text in the Description field.

- Dot Notation

  Values: text

  Internal Identifier: DotNotation

  The branch revision number, for example, 1.2.1.0.

- Dot Notation ID (Advanced)

  Values: number

  Internal Identifier: DotNotationID

  The ID assigned to a particular branch revision number. For example, if a change request was added to the current view (as opposed to inherited by the current view), its branch revision number is 1.x and its branch revision ID is 0. If a change request was branched in the current view, its branch revision ID is dependent on the revision number in the parent view and the number of IDs already assigned in the current view. For example, if a change request's revision number in the parent view is 1.7 at the time of the branch, and another change request with that same parent revision number was given the Branch Revision ID 6, this change request will also be given the Branch Revision ID 6.

- End Modified Time (Advanced)

  Values: date/time

  Internal Identifier: EndModifiedTime

  The date and time at which a revision ceased to be the tip revision. Although this field can be displayed in the upper pane, its value is always blank. This is because, at any given configuration time, the item is still the tip revision.

- Entered By

  Values: list of users, <None>

  Internal Identifier: EnteredBy

  The name of the user who created this change request.

- Entered On

  Values: date/time

  Internal Identifier: EnteredOn

  The time at which this change request was created.

- External Reference

  Values: text

  Internal Identifier: ExternalReference

Text usually used to indicate a customer or other outside source who provided the data for this change request.

- Fix

  Values: text

  Internal Identifier: Fix

  The text in the Fix field.

- Flag

  Values: No, Yes

  Internal Identifier: Flag

  A flag specifically marks/bookmarks change requests in the upper pane on your workstation.

- Flag User List (Advanced)

  Values: byte array; displayed as a bracketed series of numbers in hex format. For example, [14 00 00 00] indicates a specific user.

  Internal Identifier: FlagUserList

  Cannot be used in queries. Identifies users who have set flags on a given item.

- Folder

  Values: text

  Internal Identifier: Folder

  The name of the folder that stores the change request.

- Folder Path

  Values: text

  Internal Identifier: Folder Path (contains spaces)

  The path to the folder that stores the change request.

- Last Build Tested

  Values: list of view labels, <None>

  Internal Identifier: LastBuildTested

  The build label selected by a user to represent the last build in which a change request was tested.

- Locked By

  Values: list of users, <None>

  Internal Identifier: ExclusiveLocker

  The name of user who has exclusively locked a change request.

- Modified By

  Values: list of users, <None>

  Internal Identifier: ModifiedUserID

  The name of the user who last modified a change request.

- Modified Time

  Values: date/time

  Internal Identifier: ModifiedTime

  The time at which a change request was last modified.

- My Lock

  Values: Exclusively Locked By Me, Non-exclusively Locked By Me, Not Locked By Me

  Internal Identifier: My Lock (contains spaces)

  Indicates whether the current user has the change request locked and, if so, whether that lock is exclusive or not.

- New Revision Comment (Advanced)

  Values: text

  Internal Identifier: New Revision Comment (contains spaces)

  Internal use only. The client uses this value during the item update process. The field always appears empty if added to the upper pane.

- Non-Exclusive Lockers

  Values: text

  Internal Identifier: NonExclusiveLockers

  The names of the users who have locked the change request non-exclusively.

- Object ID

  Values: number

  Internal Identifier: ID

  Each change request is assigned an object ID when it is added to a view. When it is branched in a child view, it is assigned another object ID. The original ID belongs to the change request in the parent view.

- Parent Branch Revision (Advanced)

  Values: number

  Internal Identifier: ParentRevision

  The last number in the branch revision number before a change request branched. For example, if this number is 7, the branch revision was 1.7 at the time the change request branched (becoming 1.7.1.0, as seen in the change request's history). This number is -1 if a change request was not inherited from the parent view.

- Parent ID (Advanced)

  Values: number

  Internal Identifier: ParentObjectID

  The object ID of a change request in the parent view. The Parent ID is -1 if this view has no parent view.

- Parent Revision (Advanced)

  Values: number

  Internal Identifier: PathRevision

  The revision number at which a change request branched. For example, if this number is 8, this change request's revision number in the parent view was 8 at the time the change request branched. The history should show that revision 9 in the first revision in the current view. This number is 0 if this change request was not inherited from the parent view.

- Platform

  Values: All, MacOS, Other, Unix, Windows 2000, Windows 95, Windows 98, Windows NT, Windows XP

  Internal Identifier: Platform

The value of the Platform field.

- Priority

  Values: No, Yes

  Internal Identifier: Priority

  The value of the Priority field. Many people use repository customization to extend this field to include other values because Booleans in the application are treated as enumerated types. For example, No is 0 and Yes is 1. An administrator might change No to Not A Priority, Yes to Priority 1, and add Priorities 2 through 10.

- Read Only (Advanced)

  Values: No, Yes

  Internal Identifier: ReadOnly

  Indicates whether the change request's configuration is read-only (as in a rollback configuration of a view) and/or its behavior does not allow it to branch on modification.

- Read Status

  Values: Read, Unread

  Internal Identifier: Read Status (contains spaces)

  Indicates whether a change request is considered read or not read.

- Read Status User List

  Values: byte array; displayed as a bracketed series of numbers in hex format. For example, [14 00 00 00] indicates a specific user.

  Internal Identifier: ReadStatusUserList

  Cannot be used in queries. Identifies users for whom a given item's status is "unread".

- Resolved On

  Values: date/time

  Internal Identifier: ResolvedOn

  The time at which a change request was resolved. The resolution can be: Cannot Reproduce, As Designed, Fixed, Documented, or Is Duplicate.

- Responsibility

  Values: list of users, <None>

  Internal Identifier: Responsibility

  The name of the user who is currently responsible for a change request.

- Revision Flags (Advanced)

  Values: 0

  Internal Identifier: RevisionFlags

  Internal use only.

- Root Object ID (Advanced)

  Values: number

  Internal Identifier: RootObjectID

  The object ID of the oldest ancestor of a change request. For example, if a change request was not inherited from a parent view, the root object ID is the same as its object ID. If it was inherited from a parent view, the root object ID is the Parent ID, or the parent's Parent ID.

- Severity

  Values: High, Low, Medium

  Internal Identifier: Severity

  The value of the Severity field.

- Share State

  Values: DerivedShare, Not Shared, Root Share

  Internal Identifier: ShareState

  Indicates whether this item is shared. Not Shared means that the item is not shared. Root Share means that the item is shared and this item is the original (or root) reference. DerivedShare means that the item is shared, but this item is not the original (or root) reference.

- Short Comment

  Values: text

  Internal Identifier: ShortComment

  Stores the initial 2000 characters provided as the reason for changing a change request's properties. Additional text is stored in the Comment field.

- Status

  Values: New, Open, In Progress, Deferred, Cannot Reproduce, As Designed, Fixed, Documented, Is Duplicate, Verified Deferred, Verified Cannot Reproduce, Verified As Designed, Verified Fixed, Verified Documented, Verified Is Duplicate, Closed Deferred, Closed Cannot Reproduce, Closed As Designed, Closed Fixed, Closed Documented, Closed Is Duplicate

  Internal Identifier: Status

  The value of the Status field.

- Synopsis

  Values: text

  Internal Identifier: Synopsis

  The text in the Synopsis field.

- Test Command

  Values: text

  Internal Identifier: TestCommand

  The text in the Test Command field.

- Type

  Values: Defect, Suggestion

  Internal Identifier: Type

  The value of the Type field.

- Verified On

  Values: date/time

  Internal Identifier: VerifiedOn

  The time at which a change request was verified. The resolution can be Verified Cannot Reproduce, Verified As Designed, Verified Fixed, Verified Documented, or Verified Is Duplicate.

- Version (Advanced)

Values: number

Internal Identifier: RevisionNumber

The last number in the branch revision number. For example, if the branch revision number is 1.3.1.2, the version is 2.

- View

  Values: list of views, <None>

  Internal Identifier: ViewID

  The name of the view in which the item last branched. For example, if a change request is inherited from a parent view but is branched in the child view, the value of this field in the child view changes from the name of the parent view to the name of the child view for the revision that branched and subsequent revisions in the child view.

- Work Around

  Values: text

  Internal Identifier: WorkAround

  The text in the Work Around field.

Requirement Fields

This section lists all the requirement fields in alphabetical order.

- Am I Responsible?

  Values: No, Yes

  Internal Identifier: AmIResponsible

  Indicates whether the logged-on user is responsible for a requirement.

- Ambiguities Found

  Values: number

  Internal Identifier: AmbiguitiesFound

  Indicates the number of ambiguities found in the requirement.

- Attachment Count

  Values: number

  Internal Identifier: AttachmentCount

  The number of files attached to a requirement.

- Attachment IDs (Advanced)

  Values: byte array; displayed as a bracketed series of numbers in hex format. For example, [00 00 00 00 02 00 00 00] indicates two specific attachments.

  Internal Identifier: AttachmentIDs

  Cannot be used in queries. The ID numbers assigned to attachments. For example, the first attachment within a project is 00 00 00 00.

- Attachment Names

  Values: text containing a series of file names separated by spaces

  Internal Identifier: AttachmentNames

  The names of the files attached to a requirement.

- Children Count

  Values: number

  Internal Identifier: Children Count (contains spaces)

  The number of responses that are children of this requirement.

- ChildType

  Values: Child Requirement, Requirement

  Internal Identifier: ChildType

  Indicates whether the requirement is the root of a requirement tree or a child of another requirement.

- Comment

  Values: text

  Internal Identifier: Comment

  The initial 2000 characters provided as the reason for changing a requirement's properties are stored in the Short Comment field. This field, the Comment field, stores those 2000 characters and any additional text. Changing a requirement's properties causes the application to create a new revision.

- CommentID (Advanced)

  Values: number

  Internal Identifier: CommentID

  The ID number assigned to a revision comment. Displays -1 if no revision comment was supplied.

- Comments

  Values: text

  Internal Identifier: Comments

  Provides comments about the revised description created because of ambiguities found in the original description or for other reasons.

- Configuration Time

  Values: date/time

  Internal Identifier: ConfigurationTime

  Indicates the time to which a requirement is configured. If you configure a requirement to a specific time, this field contains that time. If you configure a requirement to a label or promotion state, this field shows either the time at which the label was created or the time at which the label associated with the promotion state was created.

- Created By

  Values: list of users, <None>

  Internal Identifier: CreatedUserID

  The name of the user who created the first revision in the view. This is the user who initiated the requirement.

- Created Time

  Values: date/time

  Internal Identifier: CreatedTime

  The time at which the first revision in the view was created.

- Deleted By

Values: list of users, <None>

Internal Identifier: DeletedUserID

The name of the user who deleted a requirement. Because deleted items do not appear in the list, this information is unavailable to users.

- Deleted Time

  Values: date/time

  Internal Identifier: DeletedTime

  The time at which a requirement was deleted. Because deleted items do not appear in the list, this information is unavailable to users.

- Description

  Values: text

  Internal Identifier: Description

  The text in the Description field.

- Disabled

  Values: No, Yes

  Internal Identifier: Disabled

  Indicates whether the requirement is disabled.

- Dot Notation

  Values: text

  Internal Identifier: DotNotation

  The branch revision number, for example, 1.2.

- End Modified Time (Advanced)

  Values: date/time

  Internal Identifier: EndModifiedTime

  The date and time at which a revision ceased to be the tip revision. Although this field can be displayed in the upper pane, its value is always blank. This is because, at any given configuration time, the item is still the tip revision.

- Expected Effort

  Values: number

  Internal Identifier: ExpectedEffort

  Indicates the expected case estimate for how long it will take to implement the requirement fully. If you are importing requirements from CaliberRM, these fields will already be filled with data based on a specific unit, such as hours or days. Otherwise, the units are arbitrary, but should be the same for the Low Effort and the High Effort fields, and should be used consistently for all requirements.

- External Reference

  Values: text

  Internal Identifier: ExternalReference

  Usually provides the name of an external customer who asked for this requirement.

- Flag

  Values: No, Yes

  Internal Identifier: Flag

A flag specifically marks/bookmarks requirements in the upper pane on your workstation.

- Flag User List (Advanced)

  Values: byte array; displayed as a bracketed series numbers in hex format. For example, [14 00 00 00] indicates a specific user.

  Internal Identifier: FlagUserList

  Cannot be used in queries. Identifies users who have set flags on a given item.

- Folder Path

  Values: text

  Internal Identifier: Folder Path (contains spaces)

  The path to the folder that stores the requirement.

- High Effort

  Values: number

  Internal Identifier: HighEffort

  Indicates the worst case estimate for how long it will take to implement the requirement fully. If you are importing requirements from CaliberRM, these fields will already be filled with data based on a specific unit, such as hours or days. Otherwise, the units are arbitrary, but should be the same for the Low Effort and the Expected Effort fields, and should be used consistently for all requirements.

- Locked By

  Values: list of users, <None>

  Internal Identifier: ExclusiveLocker

  The name of the user who has exclusively locked a requirement.

- Low Effort

  Values: number

  Internal Identifier: LowEffort

  Indicates the best case estimate for how long it will take to implement the requirement fully. If you are importing requirements from CaliberRM, these fields will already be filled with data based on a specific unit, such as hours or days. Otherwise, the units are arbitrary, but should be the same for the Expected Effort and the High Effort fields, and should be used consistently for all requirements.

- Modified By

  Values: list of users, <None>

  Internal Identifier: ModifiedUserID

  The name of the user who last modified a requirement.

- Modified Time

  Values: date/time

  Internal Identifier: ModifiedTime

  The time at which a requirement was last modified.

- My Lock

  Values: Exclusively Locked By Me, Non-exclusively Locked By Me, Not Locked By Me

  Internal Identifier: My Lock (contains spaces)

Indicates whether the current user has the requirement locked and, if so, whether that lock is exclusive or not.

- Name

  Values: text

  Internal Identifier: Name

  The name of the requirement.

- New Revision Comment (Advanced)

  Values: text

  Internal Identifier: New Revision Comment (contains spaces)

  Internal use only. A client uses this value during the item update process. The field always appears empty if added to the upper pane.

- Non-Exclusive Lockers

  Values: text

  Internal Identifier: NonExclusiveLockers

  The names of the users who have locked the requirement non-exclusively.

- Notes

  Values: text

  Internal Identifier: Notes

  Text comments on the effort levels for this requirement.

- Number

  Values: number

  Internal Identifier: RequirementNumber

  Number identifying the requirement. For example, if the Object ID is 0, the requirement number is 1.

- Object ID

  Values: number

  Internal Identifier: ID

  Each requirement is assigned an object ID when it is added to a view.

- Owner

  Values: list of users, <None>

  Internal Identifier: Owner

  Indicates who is ultimately responsible for this requirement. This field is not used by the application internally to determine ownership; the application uses the Created By field for that purpose.

- Parent Requirement ID (Advanced)

  Values: number

  Internal Identifier: ParentRequirementID

  The object ID of a requirement in the parent view. The Parent ID is -1 if this view has no parent view.

- Priority

  Values: Desirable, Essential, Unassigned, Useful

  Internal Identifier: Priority

The value of the Priority field. You can use repository customization to change the names of these values or include other values.

- Read Only (Advanced)

  Values: No, Yes

  Internal Identifier: ReadOnly

  Indicates whether the requirement's configuration is read-only (as in a rollback configuration of a view).

- Read Status

  Values: Read, Unread

  Internal Identifier: Read Status (contains spaces)

  Indicates whether a requirement is considered read or not read.

- Read Status User List

  Values: byte array; displayed as a bracketed series of numbers in hex format. For example, [14 00 00 00] indicates a specific user.

  Internal Identifier: ReadStatusUserList

  Cannot be used in queries. Identifies users for whom a given item's status is "unread".

- Responsible Count

  Values: number

  Internal Identifier: ResponsibleCount

  The number of users who are responsible for a requirement.

- Responsible IDs

  Values: byte array; displayed as a bracketed series of numbers in hex format. For example, [14 00 00 00] indicates a specific user.

  Internal Identifier: ResponsibleIDs

  Cannot be used in queries. The ID numbers assigned to the users who are responsible for the requirement.

- Responsible Names

  Values: text containing a series of user names separated by spaces

  Internal Identifier: ResponsibleNames

  The names of the users responsible for this requirement.

- Reviewed By

  Values: byte array

  Internal Identifier: ReviewedByIDs

  Cannot be used in queries. Should not be used at all.

- Revised Description

  Values: text

  Internal Identifier: RevisedDescription

  Provides a new, revised description because of ambiguities found in the original description or for other reasons.

- Revision Flags (Advanced)

  Values: 0

Internal Identifier: RevisionFlags

Internal use only.

- Share State

  Values: DerivedShare, Not Shared, Root Share

  Internal Identifier: ShareState

  Indicates whether this item is shared. Not Shared means that the item is not shared. Root Share means that the item is shared and this item is the original (or root) reference. DerivedShare means that the item is shared, but this item is not the original (or root) reference.

- Short Comment

  Values: text

  Internal Identifier: ShortComment

  Stores the initial 2000 characters provided as the reason for changing a requirement's properties. Additional text is stored in the Comment field.

- Status

  Values: Accepted, Approved, Complete, Deferred, Draft, Pending, ReadyForCCB, Rejected, Review, Submitted

  Internal Identifier: Status

  Indicates the status of this requirement.

- Type

  Values: Business Requirement, Business Specification, Hardware Requirement, Hardware Specification, Human Resources, Information Technology, Software Requirement, Software Specification

  Internal Identifier: Type

  Indicates the type of requirement.

- Version (Advanced)

  Values: number

  Internal Identifier: RevisionNumber

  The last number in the branch revision number. For example, if the branch revision number is 1.2, the version is 2.

Task Fields

This section lists all the task fields in alphabetical order.

- Actual Finish

  Values: date/time

  Internal Identifier: StTaskActualFinish

  The actual finish date for a task.

- Actual Hours

  Values: number

  Internal Identifier: StTaskActualHours

  The number of hours spent completing the task.

- Actual Start

  Values: date/time

  Internal Identifier: StTaskActualStart

  The actual start date for a task.

- Attachment Count

  Values: number

  Internal Identifier: AttachmentCount

  The number of files attached to a task.

- Attachment IDs (Advanced)

  Values: byte array; displayed as a bracketed series of numbers in hex format. For example, [00 00 00 00 02 00 00 00] indicates two specific attachments.

  Internal Identifier: AttachmentIDs

  Cannot be used in queries. The ID numbers assigned to attachments. For example, the first attachment within a project is 00 00 00 00.

- Attachment Names

  Values: text containing a series of file names separated by spaces

  Internal Identifier: AttachmentNames

  The names of the files attached to a task.

- Attention Notes

  Values: text

  Internal Identifier: StTaskAttentionNotes

  The text in the “Needs Attention” note.

- Children Count

  Values: number

  Internal Identifier: Children Count (contains spaces)

  The number of responses that are subtasks of this task.

- Comment

  Values: text

  Internal Identifier: Comment

  The initial 2000 characters provided as the reason for changing a change request’s properties are stored in the Short Comment field. This field, the Comment field, stores those 2000 characters and any additional text. Changing a task’s properties causes the application to create a new revision.

- CommentID (Advanced)

  Values: number

  Internal Identifier: CommentID

  The ID number assigned to revision comment. Displays -1 if no revision comment was supplied.

- Configuration Time

  Values: date/time

  Internal Identifier: ConfigurationTime

Indicates the time to which a task is configured. If you configure a task to a specific time, this field contains that time. If you configure a task to a label or promotion state, this field shows either the time at which the label was created or the time at which the label associated with the promotion state was created.

- Constraint Date

  Values: date/time

  Internal Identifier: StTaskConstraintDate

  A task's constraint date from MS Project.

- Constraint Type

  Values: As Late As Possible, As Soon As Possible, Finish No Earlier Than, Finish No Later Than, Must Finish On, Must Start On, Start No Earlier Than, Start No Later Than

  Internal Identifier: StTaskConstraintType

  A task's constraint type from MS Project.

- Created By

  Values: list of users, <None>

  Internal Identifier: CreatedUserID

  The name of the user who created the first revision in the view. This is the user who initiated the task.

- Created Time

  Values: date/time

  Internal Identifier: CreatedTime

  The time at which the first revision in the view was created.

- Deleted By

  Values: list of users, <None>

  Internal Identifier: DeletedUserID

  The name of the user who deleted a task. Because deleted tasks do not appear in the list, this information is unavailable to users.

- Deleted Time

  Values: date/time

  Internal Identifier: DeletedTime

  The time at which a task was deleted. Because deleted tasks do not appear in the list, this information is unavailable to users.

- Dot Notation

  Values: text

  Internal Identifier: DotNotation

  The branch revision number, for example, 1.2.

- End Modified Time (Advanced)

  Values: date/time

  Internal Identifier: EndModifiedTime

  The date and time at which a revision ceased to be the tip revision. Although this field can be displayed in the upper pane, its value is always blank. This is because, at any given configuration time, the item is still the tip revision.

- Estimated Finish

  Values: date/time

  Internal Identifier: StTaskEstimatedFinish

  The estimated finish date for a task.

- Estimated Finish Variance

  Values: date/time

  Internal Identifier: StTaskEstimatedFinishVariance

  The difference between the estimated and the actual finish date for a task.

- Estimated Hours

  Values: number

  Internal Identifier: StTaskEstimatedHours

  The number of hours spent completing the task.

- Estimated Hours Variance

  Values: number

  Internal Identifier: StTaskEstimatedHoursVariance

  The difference between the estimated and the actual number of hours spent completing the task.

- Estimated Start

  Values: date/time

  Internal Identifier: StTaskEstimatedStart

  The estimated start date for a task.

- Estimated Start Variance

  Values: date/time

  Internal Identifier: StTaskEstimatedStartVariance

  The difference between the estimated and the actual start date for a task.

- Flag

  Values: No, Yes

  Internal Identifier: Flag

  A flag specifically marks/bookmarks tasks in the upper pane on your workstation.

- Flag User List (Advanced)

  Values: byte array; displayed as a bracketed series of numbers in hex format. For example, [14 00 00 00] indicates a specific user.

  Internal Identifier: FlagUserList

  Cannot be used in queries. Identifies users who have set flags on a given item.

- Folder Path

  Values: text

  Internal Identifier: Folder Path (contains spaces)

  The path to the folder that stores the task.

- Is My Task?

  Values: No, Yes

  Internal Identifier: Is My Task? (contains spaces)

Indicates whether the logged on user is responsible for a task.

- Is Replicated

  Values: No, Yes

  Internal Identifier: Is Replicated (contains spaces)

  Indicates whether the task is from MS Project task.

- Last MS Project Update

  Values: date/time

  Internal Identifier: StTaskMSProjectLastUpdate

  The date that a task was last updated from MS Project.

- Last Work/Dependency Update

  Values: date/time

  Internal Identifier: StWorkDependencyLastUpdate

  The last time that a work record or a dependency (task successor or predecessor) was added, edited, or deleted. This field is for use with MS Project.

- Locked By

  Values: list of users, <None>

  Internal Identifier: ExclusiveLocker

  The name of user who has exclusively locked a task.

- Milestone

  Values: No, Yes

  Internal Identifier: StTaskMilestone

  Indicates whether a task represents a milestone. In MS Project, the definition for a milestone is a task of zero time length. It serves as a heading for one or more tasks to which a time length has been assigned.

  In the application, a task has a milestone check box. After work is assigned to a task, it is no longer a milestone.

- Modified By

  Values: list of users, <None>

  Internal Identifier: ModifiedUserID

  The name of the user who last modified a task.

- Modified Time

  Values: date/time

  Internal Identifier: ModifiedTime

  The time at which a task was last modified.

- MS Project File Name (Advanced)

  Values: text

  Internal Identifier: StTaskMSProjectFileName

  The name of the MS project file from which a task was exported.

- MS Task GUID (Advanced)

  Values: text

  Internal Identifier: StTaskGUID

The GUID for a task in MS Project.

- MS Task Unique ID (Advanced)

  Values: number

  Internal Identifier: StTaskUniqueID

  The unique ID for a task in MS Project.

- MS WBS Code (Advanced)

  Values: text

  Internal Identifier: StTaskWBSCode

  A task's WBS code from MS Project

- My Lock

  Values: Exclusively Locked By Me, Non-exclusively Locked By Me, Not Locked By Me

  Internal Identifier: My Lock (contains spaces)

  Indicates whether the current user has the task locked and, if so, whether that lock is exclusive or not.

- Needs Attention

  Values: No, Yes

  Internal Identifier: StTaskNeedsAttention

  Indicates that the check box for Needs Attention has been selected.

- New Revision Comment (Advanced)

  Values: text

  Internal Identifier: New Revision Comment (contains spaces)

  Internal use only. The client uses this value during the item update process. The field always appears empty if added to the upper pane.

- Non-Exclusive Lockers

  Values: text

  Internal Identifier: NonExclusiveLockers

  The names of the users who have locked the task non-exclusively.

- Notes

  Values: text

  Internal Identifier: StTaskNotes

  The text of the note that accompanies the Needs Attention field.

- Object ID

  Values: number

  Internal Identifier: ID

  Each task is assigned an object ID when it is added to a view.

- Parent Task ID (Advanced)

  Values: number

  Internal Identifier: StTaskParentID

  The object ID of a task in the parent view. The Parent ID is -1 if this view has no parent view.

- Percent Complete

  Values: number

  Internal Identifier: StTaskPercentComplete

  A percentage indicating how much of a task has been completed.

- Priority

  Values: Do Not Level, High, Higher, Highest, Low, Lower, Lowest, Medium, Very High, Very Low

  Internal Identifier: StTaskPriority

  Indicates the priority given to a task. These priorities are identical to those in MS Project.

- Read Only (Advanced)

  Values: No, Yes

  Internal Identifier: ReadOnly

  Indicates whether the task's configuration is read-only (as in a rollback configuration of a view).

- Read Status

  Values: Read, Unread

  Internal Identifier: Read Status (contains spaces)

  Indicates whether a task is considered read or not read.

- Read Status User List

  Values: byte array; displayed as a bracketed series of numbers in hex format. For example, [14 00 00 00] indicates a specific user.

  Internal Identifier: ReadStatusUserList

  Cannot be used in queries. Identifies users for whom a given item's status is "unread".

- Resource Count

  Values: number

  Internal Identifier: StTaskResourceCount

  The number of users listed as resources for a task.

- Resource IDs (Advanced)

  Values: byte array; displayed as a bracketed series of numbers in hex format. For example, [14 00 00 00] indicates a specific user.

  Internal Identifier: StTaskResourceIDs

  Cannot be used in queries. The ID numbers assigned to the users who are this task's resources.

- Resource Names

  Values: text containing a series of user names separated by spaces

  Internal Identifier: StTaskResourceNames

  The names of the users who are this task's resources.

- Responsibility

  Values: list of users, <None>

  Internal Identifier: StTaskResponsibility

The name of the user who is currently responsible for the task.

- Revision Flags (Advanced)

  Values: 0

  Internal Identifier: RevisionFlags

  Internal use only.

- Share State

  Values: DerivedShare, Not Shared, Root Share

  Internal Identifier: ShareState

  Indicates whether this item is shared. Not Shared means that the item is not shared. Root Share means that the item is shared and this item is the original (or root) reference. DerivedShare means that the item is shared, but this item is not the original (or root) reference.

- Short Comment

  Values: text

  Internal Identifier: ShortComment

  Stores the initial 2000 characters provided as the reason for changing a task's properties. Additional characters are stored in the Comment field.

- Status

  Values: Closed, Finish, Hold, In Progress, Pending, Ready To Start

  Internal Identifier: StTaskStatus

  Indicates the status of the task.

- Task Duration

  Values: number

  Internal Identifier: StTaskDuration

  The number of hours during which any user is working on a task. For example if two people will work eight hours on a task, the duration is eight hours if they work at the same time or a maximum of 16 hours if they do the work on different days.

- Task Name

  Values: text

  Internal Identifier: StTaskName

  The name of a task.

- Task Number

  Values: number

  Internal Identifier: StTaskNumber

  The number assigned to a task. For example, if the Object ID is 0, the task number is 1.

- Task Origin

  Values: MSProject, StarTeam

  Internal Identifier: StTaskOrigin

  Indicates whether the task was created in the application or exported to the application from Microsoft Project.

- Task Type

  Values: Fixed Duration, Fixed Units, Fixed Work

Internal Identifier: StTaskType

A task's type in MS Project.

- Version (Advanced)

  Values: number

  Internal Identifier: RevisionNumber

  The last number in the branch revision number. For example, if the branch revision number is 1.2, the version is 2.

- Work Record Count

  Values: number

  Internal Identifier: WorkRecCount

  The number of work records currently added to a task.

Topic Fields

This section lists all the topic fields in alphabetical order.

- Am I Recipient?

  Values: No, Yes

  Internal Identifier: Am I Recipient? (contains spaces)

  Indicates whether the logged on user is a recipient of a topic.

- Attachment Count

  Values: number

  Internal Identifier: AttachmentCount

  The number of files attached to a topic.

- Attachment IDs (Advanced)

  Values: byte array; displayed as a bracketed series of numbers in hex format. For example, [00 00 00 00 02 00 00 00] indicates two specific attachments.

  Internal Identifier: AttachmentIDs

  Cannot be used in queries. The ID numbers assigned to attachments. For example, the first attachment within a project is 00 00 00 00.

- Attachment Names

  Values: text containing a series of file names separated by spaces

  Internal Identifier: AttachmentNames

  The names of the files attached to a topic.

- Children Count

  Values: number

  Internal Identifier: Children Count (contains spaces)

  The number of responses that are children of this topic.

- Comment

  Values: text

  Internal Identifier: Comment

  The initial 2000 characters provided as the reason for changing a topic's properties are stored in the Short Comment field. This field, the Comment field, stores those

2000 characters and any additional text. Changing a topic's properties causes the application to create a new revision.

- CommentID (Advanced)

  Values: number

  Internal Identifier: CommentID

  The ID number assigned to revision comment. Displays -1 if no revision comment was supplied.

- Configuration Time

  Values: date/time

  Internal Identifier: ConfigurationTime

  Indicates the time to which a topic is configured. If you configure a topic to a specific time, this field contains that time. If you configure a topic to a label or promotion state, this field shows either the time at which the label was created or the time at which the label associated with the promotion state was created.

- Content

  Values: text

  Internal Identifier: Description

  The text of a topic.

- Created By

  Values: list of users, <None>

  Internal Identifier: CreatedUserID

  The name of the user who created the first revision in the view. This is the user who initiated the topic.

- Created Time

  Values: date/time

  Internal Identifier: CreatedTime

  The time at which the first revision in the view was created.

- Deleted By

  Values: list of users, <None>

  Internal Identifier: DeletedUserID

  The name of the user who deleted a topic. Because deleted items do not appear in the list, this information is unavailable to users.

- Deleted Time

  Values: date/time

  Internal Identifier: DeletedTime

  The time at which a topic was deleted. Because deleted items do not appear in the list, this information is unavailable to users.

- Dot Notation

  Values: text

  Internal Identifier: DotNotation

  The branch revision number, for example, 1.2.

- End Modified Time (Advanced)

  Values: date/time

Internal Identifier: EndModifiedTime

The date and time at which a revision ceased to be the tip revision. Although this field can be displayed in the upper pane, its value is always blank. This is because, at any given configuration time, the item is still the tip revision.

- Flag

  Values: No, Yes

  Internal Identifier: Flag

  A flag specifically marks/bookmarks topics and/or responses in the upper pane on your workstation.

- Flag User List (Advanced)

  Values: byte array; displayed as a bracketed series of numbers in hex format. For example, [14 00 00 00] indicates a specific user.

  Internal Identifier: FlagUserList

  Cannot be used in queries. Identifies users who have set flags on a given item.

- Folder Path

  Values: text

  Internal Identifier: Folder Path (contains spaces)

  The path to the folder that stores the topic.

- Locked By

  Values: list of users, <None>

  Internal Identifier: ExclusiveLocker

  The name of user who has exclusively locked a topic.

- Modified By

  Values: list of users, <None>

  Internal Identifier: ModifiedUserID

  The name of the user who last modified a topic.

- Modified Time

  Values: date/time

  Internal Identifier: ModifiedTime

  The time at which a topic was last modified.

- My Lock

  Values: Exclusively Locked By Me, Non-exclusively Locked By Me, Not Locked By Me

  Internal Identifier: My Lock (contains spaces)

  Indicates whether the current user has the topic locked and, if so, whether that lock is exclusive or not.

- New Revision Comment (Advanced)

  Values: text

  Internal Identifier: New Revision Comment (contains spaces)

  Internal use only. the client uses this value during the item update process. The field always appears empty if added to the upper pane.

- Non-Exclusive Lockers

Values: text

Internal Identifier: NonExclusiveLockers

The names of the users who have locked the topic non-exclusively.

- Object ID

  Values: number

  Internal Identifier: ID

  Each topic is assigned an object ID when it is added to a view.

- Parent Topic ID (Advanced)

  Values: number

  Internal Identifier: ParentTopicID

  The object ID of a topic in the parent view. The Parent ID is -1 if this view has no parent view.

- Priority

  Values: High, Low, Normal

  Internal Identifier: Priority

  The value of the Priority field. You can use repository customization to change the names of these values or include other values.

- Read Only (Advanced)

  Values: No, Yes

  Internal Identifier: ReadOnly

  Indicates whether the topic's configuration is read-only (as in a rollback configuration of a view).

- Read Status

  Values: Read, Unread

  Internal Identifier: Read Status (contains spaces)

  Indicates whether a topic is considered read or not read.

- Read Status User List

  Values: byte array; displayed as a bracketed series of numbers in hex format. For example, [14 00 00 00] indicates a specific user.

  Internal Identifier: ReadStatusUserList

  Cannot be used in queries. Identifies users for whom a given item's status is "unread".

- Recipient Count

  Values: number

  Internal Identifier: RecipientCount

  The number of recipients to whom a topic is addressed.

- Recipient IDs

  Values: byte array; displayed as a bracketed series of numbers in hex format. For example, [14 00 00 00] indicates a specific user.

  Internal Identifier: RecipientIDs

  Cannot be used in queries. The ID numbers assigned to the users who are recipients (people to be notified about this topic).

- Recipient Names

  Values: text containing a series of users names separated by spaces

  Internal Identifier: RecipientNames

  The names of the recipients designated for notification about this topic.

- Revision Flags (Advanced)

  Values: 0

  Internal Identifier: RevisionFlags

  Internal use only.

- Share State

  Values: DerivedShare, Not Shared, Root Share

  Internal Identifier: ShareState

  Indicates whether this item is shared. Not Shared means that the item is not shared. Root Share means that the item is shared and this item is the original (or root) reference. DerivedShare means that the item is shared, but this item is not the original (or root) reference.

- Short Comment

  Values: text

  Internal Identifier: ShortComment

  Stores the initial 2000 characters provided as the reason for changing a topic's properties. Additional text is stored in the Comment field.

- Status

  Values: Active, Inactive

  Internal Identifier: Status

  Indicates the status of this topic.

- Title

  Values: text

  Internal Identifier: Title

  The text of the Title field.

- Topic Number

  Values: number

  Internal Identifier: TopicNumber

  The number assigned to a topic. For example, if the Object ID is 0, the topic number is 1.

- Type

  Values: Response, Topic

  Internal Identifier: Type

  Indicates whether the item is a topic (root of a topic tree) or a response (branch of a topic tree).

- Version (Advanced)

  Values: number

  Internal Identifier: RevisionNumber

The last number in the branch revision number. For example, if the branch revision number is 1.2, the version is 2.

Audit Log Fields

This section lists all the audit entry fields in alphabetical order.

- Class Name 1

  Values: text

  Internal Identifier: Class Name 1 (contains spaces)

  The name of the class of items, such as Label, Promotion State, Folder, File, Change Request, Topic, or Task.

- Class Name 2

  Values: text

  Internal Identifier: Class Name 2 (contains spaces)

  The name of the class of items, such as Folder, File, Change Request, Label, Topic, or Task.

- Class Name 3

  Values: text

  Internal Identifier: Class Name 3 (contains spaces)

  The name of the class of items, such as Folder, File, Change Request, Label, Topic, or Task.

- Created By

  Values: list of users, <None>

  Internal Identifier: CreatedUserID

  Always empty because the audit entry is created by the system.

- Created Time

  Values: date/time

  Internal Identifier: CreatedTime

  The time at which this entry was created.

- Deleted By

  Values: list of users, <None>

  Internal Identifier: DeletedUserID

  The name of the user who deleted an audit entry. Because deleted entries do not appear in the list, this information is unavailable to users.

- Deleted Time

  Values: date/time

  Internal Identifier: DeletedTime

  The time at which an audit entry was deleted. Because deleted entries do not appear in the list, this information is unavailable to users.

- Event

  Values: Added, Branched, Comment Edited, Created, Deleted, Edited, Item Overwritten, Label Attached, Label Created, Label Deleted, Label Detatched, Label Frozen, Label Modified, Label Moved, Label Unfrozen, Lock Broken,

Locked, Modified, Moved From, Moved To, Promotion Model Modified, Promotion State Modified, Shared, Unlocked, Vault Converted

Internal Identifier: EventID

The name of the operation being recorded.

- Folder

  Values: text

  Internal Identifier: Folder

  The name of the folder that stores the audit entry.

- Folder Path

  Values: text

  Internal Identifier: Folder Path (contains spaces)

  The path to the folder that stores the audit entry.

- Folder VMID (Advanced)

  Values: number

  Internal Identifier: FolderVMID

  The ID assigned to the folder that stores the item.

- Item 1

  Values: text

  Internal Identifier: Item 1 (contains spaces)

  Indicates what class 1 item received the audited operation. This can be the name of a file or task, the number of a change request or requirement, or the title of a topic.

- Item 1 Info

  Values: text

  Internal Identifier: Info

  Provides the revision number in dot notation for the class 1 item, if it is revisionable. For example, a label can be a class 1 item and it does not have revisions.

- Item 2

  Values: text

  Internal Identifier: Item 2 (contains spaces)

  Indicates what class 2 item received the audited operation. For example, if a label was attached to a file, the class 1 item is the label and the class 2 item is the file.

- Item 2 Info

  Values: text

  Internal Identifier: Info2

  Provides the revision number in dot notation for the class 2 item, if it is revisionable. For example, a label can be a class 2 item and it does not have revisions.

- Item 3

  Values: text

  Internal Identifier: Item 3 (contains spaces)

  Indicates what class 3 item received the audited operation. For example, if a label was moved from one revision to a file to another, the class 1 item is the label, the class 2 item is the revision of the file that was initially , and the class 3 item is the final revision of the file.

- Item 3 Info

  Values: text

  Internal Identifier: Info3

  Provides the revision number in dot notation for the class 3 item, if it is revisionable. For example, a label can be a class 1 item and it does not have revisions.

- Modified By

  Values: list of users, <None>

  Internal Identifier: ModifiedUserID

  Does not apply to audit entries.

- Modified Time

  Values: date/time

  Internal Identifier: ModifiedTime

  Does not apply to audit entries.

- Object ID

  Values: number

  Internal Identifier: ID

  Each audit entry is assigned an object ID when it is added to a view.

- Project

  Values: list of projects in this server configuration, <None>

  Internal Identifier: ProjectID

  The name of the project in which an audit entry was recorded.

- Target 1 Class ID (Advanced)

  Values: number

  Internal Identifier: Target 1 Class ID (contains spaces)

  The ID number assigned to class 1 items or a -1 if there is no ID.

- Target 1 Object ID (Advanced)

  Values: number

  Internal Identifier: Target 1 Object ID (contains spaces)

  The object ID for the class 1 item that received the audited operation or a -1 if there is no ID.

- Target 1 Revision Time

  Values: date/time

  Internal Identifier: Target 1 Revision Time (contains spaces)

  The time at which the last revision was made to the class 1 item that received the audit operation.

- Target 2 Class ID (Advanced)

  Values: number

  Internal Identifier: Target 2 Class ID (contains spaces)

  The ID number assigned to class 2 items or a -1 if there is no ID.

- Target 2 Object ID (Advanced)

  Values: number

Internal Identifier: Target 2 Object ID (contains spaces)

The object ID for the class 2 item that received the audited operation or a -1 if there is no ID.

- Target 2 Revision Time

  Values: date/time

  Internal Identifier: Target 2 Revision Time (contains spaces)

  The time at which the last revision was made to the class 2 item that received the audit operation.

- Target 3 Class ID (Advanced)

  Values: number

  Internal Identifier: Target 3 Class ID (contains spaces)

  The ID number assigned to class 3 items or a -1 if there is no ID.

- Target 3 Object ID (Advanced)

  Values: number

  Internal Identifier: Target 3 Object ID (contains spaces)

  The object ID for the class 3 item that received the audited operation or a -1 if there is no ID.

- Target 3 Revision Time

  Values: date/time

  Internal Identifier: Target 3 Revision Time (contains spaces)

  The time at which the last revision was made to the class 3 item that received the audit operation.

- User

  Values: list of users, <None>

  Internal Identifier: UserID

  The name of the user who performed the recorded operation.

- View

  Values: list of views, <None>

  Internal Identifier: ViewID

  The name of the view in which an audit entry was recorded.

Chapter

7

Using Visual Merge and Visual Diff

Visual Merge and Visual Diff are utilities that help you locate and manage changes between files and file revisions. Both utilities can be run from:

- The Windows Start menu
- The command line
- Within the application (Windows and Cross-Platform client when installed on a Windows operating system)

Visual Diff is a two-way text comparison utility that compares text files, two different revisions of text files, binary files and folder contents. It compares the two side-by-side. Visual Diff automatically starts when you compare file revisions (see “Comparing File Contents” on page 140 for more information).

Visual Merge is a three-way merge utility that can merge text files or binary files. As with Visual Diff, Visual Merge automatically starts when you check a file in or out that has a status of Merge. It is available on the Windows client and on the Cross-Platform client when it is installed on a Windows operating system.

As an example of how Visual Merge works, suppose two users check out the same text file from the server. They both then modify that file on their workstations. Three similar files now exist:

- The revision originally checked out to both workstations
- The file as modified by user No. 1
- The file as modified by user No. 2

When user No. 1 checks in the new revision, the status of user No. 2’s file becomes Merge, alerting user No. 2 to the problem. When user No. 2 checks in his or her file, Visual Merge compares user No. 1’s work to the original file in one pane, user No. 2’s work to the original file in the second pane, and suggests a merged file in the third pane. By suggesting this merged file, no work needs to be lost.

Comparing the Visual Merge and Visual Diff Utilities

The following detailed example illustrates the differences between these two utilities.

Revision 1 of an autoexec.bat file contained the following lines:

```
set temp=c:\windows\temp
path=c:\;c:\windows\command;c:\windows;
c:\dos\doskey
c:\mouse\mouse
C:\TOOLKIT\GUARD
set pvxfid0=T111
```

Two users check it out and modify it. Revision 2 is the autoexec.bat file created by the first user to check the file back in. It now has one additional line about qbackup:

```
set temp=c:\windows\temp
path=c:\;c:\windows\command;c:\windows;c:\dos
c:\dos\doskey
c:\mouse\mouse
C:\TOOLKIT\GUARD
set pvxfid0=T111
set qbackup=c:\qbackup\data
```

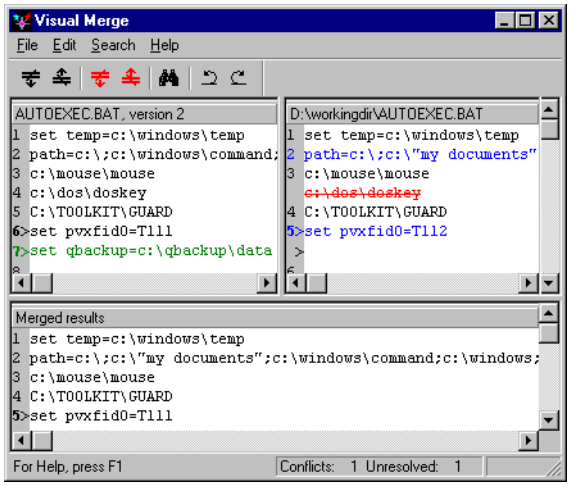
As the second user attempts to check in the file, the status of the file in his or her working folder has changed from Current to Merge. The application indicates that the file has changed since the second user checked out a copy of it.

This copy of the autoexec.bat file, the second user's working file, contains the following:

```
set temp=c:\windows\temp
path=c:\;c:\"my documents";
   c:\windows\command;c:\windows;c:\dos
c:\mouse\mouse
C:\TOOLKIT\GUARD
set pvxfid0=T112
```

The doskey line has been deleted and both the path and the set pvxfid0 lines have been changed.

After the second user checks in the file and the application performs a merge, Visual Merge displays three panes within the Visual Merge window:

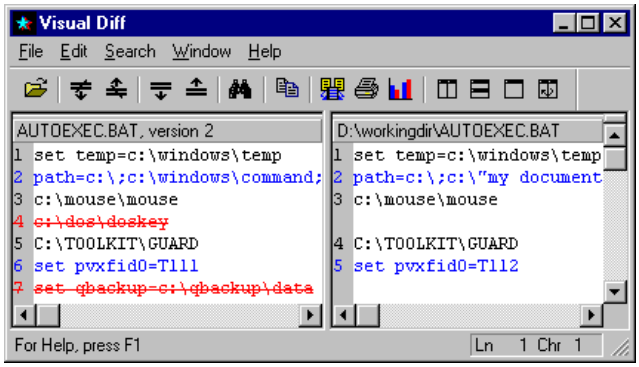


The upper left pane shows Revision 2 (the first user's work) compared to Revision 1. Line 7 is green to indicate that it has been added.

The upper right pane compares the second user's working file (that has not been checked in yet) to Revision 1. The doskey line is red and struck through because it was deleted. The path and set pvxfid0 lines are blue because they have changed.

The the bottom pane, Merged Results, shows an attempt to resolve these changes. Because it could not resolve line 5 of the lower pane, the line to set pvxfid0 to either 111 or 112, you must resolve it.

If you use Visual Diff to compare Revision 2 and the remaining working copy, it determines how they differ from each other. Because Revision 2 has neither a `doskey` or a `qbackup` line, Visual Diff assumes these lines were both deleted from the second working copy. Visual Diff identifies, as did Visual Merge, that the `path` and the `set pvxfid0` lines have been changed. However, Visual Diff doesn't identify any lines as added because all the lines in the remaining working copy were in Revision 2.



Using Visual Merge

Visual Merge can be used from the Windows client or from the Cross-Platform client when it is installed on a Windows operating system.

Visual Merge can be used:

- As a stand-alone utility with a graphical user interface
- As a stand-alone utility at the command line
- When you are checking a file in or out that has the Merge status

You can use Visual Merge with either text or binary files.

If you force the check-in of a file with the Merge status, the changes made in all the revisions of this file since you checked it out will be overwritten.

Using the Stand-alone Utility

To access Visual Merge as a stand-alone utility with a graphical user interface:

1 Do one of the following:

- Double-click the Visual Merge icon in the application program group.
- Select Start > Programs > StarTeam > StarTeam client_name > Visual Merge.

The *Visual Merge Info* dialog appears.

2 Visual Merge is a three-way comparison utility and merges all lines that have no conflicts. Conflicts are defined as lines that have been changed in both files (or file revisions) so Visual Merge cannot determine automatically what changes to retain in the merged file.

You must provide the paths to all three files:

a In the Common Ancestor File text box, enter or browse for the common ancestor of the other two files, the one to which each of them will be compared.

b In the First File text box, enter or browse for one of the two files.

c In the Second File text box, enter or browse for the other of the two files.

3 In the Results File text box, enter or browse for the file that will store the merged file that results from the three-way comparison and your editing.

4 Click OK to start the merging process.

To access Visual Merge as a stand-alone utility from the command line:

- At the command line, use the following syntax: l

vismerge	[**-am** |**-amt** |**-at**] [**-iw**] *ancestor_file_path first_file_path second_file_path* [*result_file_path*] [*ancestor_caption*] [*first_file_caption*] [*second_file_caption*] If you do not use -am, -amt, or -at, the Visual Merge window opens and you must perform all merging. Nothing is automatic.
-am	Causes Visual Merge to automatically merge all lines without conflicts; then the Visual Merge window opens so you can check the result file.
-amt	Causes Visual Merge to automatically merge all lines without conflicts and then terminates if there are no conflicts. The Visual Merge window opens only if there are conflicts in the result file.
-at	Causes Visual Merge to automatically merge only non-conflicting changes. The Visual Merge window never opens. If there are no conflicts, the merged result is saved to the result file. Otherwise, the command terminates.
-iw	Causes Visual Merge to ignore whitespace. For example if one file has two tabs where the other file has eight spaces, the tabs and spaces are considered equivalent.
ancestor_ file_path	The path to the common file upon which both of the files to be merged is based.
first_ file_path	The first of two files to be used in the merge.
second_ file_path	The second of two files to be used in the merge
result_ file_path	(Optional) The path to the file that will store the merged results. When not used, the merged text file, if one is created, becomes standard output. That means you can see the merged text in the DOS window or redirect that text to another file, for example: `vismerge -am common.txt file1.txt file2.txt > result.txt`
ancestor_ caption	Title for the lower window in Visual Merge (preceded with “Derived from”). Using the name of the ancestor file as the caption is recommended.
first_file_ caption	Title for the upper left window in Visual Merge. Using the name of the file specified by the first_file_path option is recommended.
second_file_ caption	Title for the upper right window in Visual Merge. Using the name of the file specified by the second_file_path option is recommended.

Using Visual Merge

Whenever you check in or check out a text file that has Merge status, the application asks if you want to merge that file. If you select Yes, the application starts Visual Merge. Select No or Cancel to stop the check-in or check-out operation.

A personal option controls whether you:

- Always see the Visual Merge main window so that you can edit the merged file only when there are conflicts.
-
- Automatically check the merged file (created by Visual Merge) in or out, unless conflict resolutions are required. Then you see the Visual Merge main window.

This option applies to text files only. For binary files, Visual Merge identifies conflicts.

To set Visual Merge personal options:

1. Select Tools > Personal Options from the menu bar.
2. In the resulting *Personal Options* dialog, select the File tab.
3. Select the context Visual Merge In Case Of Conflicts option button to be notified only when there are conflicts in a merged file.
4. Click OK.

Using the Main Window to Merge Text Files

The Visual Merge main window consists of a menu bar, a tool bar, and three panes. The panes display the two files to be merged and a suggested merged file. The two files to be merged appear in the upper left and right panes. Each pane shows how its file differs from the revision it is based on.

The differences are shown in color. The default colors are as follows:

- Black. Matching blocks of text are displayed in black.
- Red. Deleted blocks of text appear in red and struck through.
- Green. Inserted blocks of text appear in green.
- Blue. Changed blocks of text appear in blue. Text that is similar but not identical is considered to be changed.

Note If you change the colors in Visual Merge, Visual Merge checks the background color for the system to ensure that no color conflicts exist. If the background color is the same as one of your colors, Visual Merge uses the system text color instead.

The suggested merged file appears in the lower pane and uses no identifying colors. This is the only pane you can edit. From the upper two panes, you can only copy text.

If you right-click on a changed, new, or deleted block of text in one of the upper panes, Visual Merge displays a context menu.

Clicking the menu options alters the text in the lower pane. For example, suppose you have a block of changed text (blue, by default). This block appears in blue in both of the upper panes, but they are not identical blocks. One of the two blocks also appears in the lower pane (Visual Merge deduces which change is the desired one.)

If you right-click one of the changed blocks and select Apply Both Changes from the context menu, your block of text and the corresponding text in the other upper pane are both displayed in the lower pane. One of the changes was already there, but now you have both. If you right-clicked the changed block and selected Apply Change from the context menu, your block of text would have been added to the lower pane, replacing the text that was already there. Depending on what pane you clicked, your block of text may be replacing itself or replacing the corresponding text from the other upper pane. Had you selected Remove Change, the selected block of text would have disappeared from the lower pane.

The behavior is slightly different for added and deleted blocks of text, but Visual Merge still deduces the most logical choice. For example, if you right-click a deleted block of text (red, by default) and click Remove Change, the deleted block is inserted into the lower pane. You are removing the deletion.

Changing File Encodings

By default, Visual Merge assumes that the text files used in the three-way-merge and the result file all have the same encoding as the default code page for the operating

system. When the files have different encodings, comparing them is difficult if not impossible until you specify the correct encoding for each file.

To change the encoding for a file used by or created by Visual Merge:

1. Select Edit > Encoding from the menu bar. The *File Encoding* dialog appears.
2. Select the option button for the file whose encoding needs to be changed.
3. Select the correct encoding from the Encoding For Selected File list box.
4. Repeat steps 2 and 3 for other files with the wrong encoding.
5. Click OK to return to Visual Merge.

Searching Text Files

The Visual Merge Find command locates the next instance of the search string in the pane where you have placed the cursor. Visual Merge also keeps all three panes synchronized so that the same string (or its approximate former location) appears in all three panes.

To find text:

1. Do one of the following:
   - Select Search > Find from the menu bar.
   - Click the Find icon on the toolbar.

   The *Find* dialog appears.
2. Enter the string to be located in the Find What text box.
3. If you want to locate the text exactly as you entered it with regard to case, select the Match Case check box.
4. If necessary, specify a direction for the search.

   Visual Merge remembers the search string until you enter another one or close the application. Change panes to search in another pane.
5. Click Find Next.
6. To locate the next instance of the text, select Search > Find Next.

Note You can search for the next or previous set of differences using either the Search menu from the Visual Merge menu bar or the toolbar buttons. Visual Merge finds the lines and positions them at the top of each pane.

To find the next or previous difference, do one of the following:

- Select Search > Next Difference from the menu bar.
- Click Next Difference icon on the toolbar.
- Select Search > Previous Difference from the menu bar.
- Click Previous Difference icon on the toolbar.

In addition, you can search for the next or previous set of conflicts, which are lines that have changed and need your intervention to prevent data from being lost.

To find the next or previous conflict, do one of the following:

- Select Search > Next Conflict from the menu bar.
- Click Next Conflict icon on the toolbar.
- Select Search > Previous Conflict from the menu bar.
- Click Previous Conflict icon on the toolbar.

Exiting Visual Merge

In you are using Visual Merge to merge files as part of a check-in or check-out operation, there are steps to perform as you exit. After you edit the merged file, perhaps with the help of a user whose changes are in danger of being lost, you must decide whether to save the merged file. If you are checking in the file, you must also decide whether to check in the merged file or only replace the working file.

To exit Visual Merge:

1. Select File > Exit from the menu bar.

   Visual Merge asks if you want to save the file. In the same message dialog, Visual Merge also notifies you if the file has unresolved conflicts (if there are any).

2. Do one of the following:

   - Click Yes to exit Visual Merge and start the process of saving the merged file as your working file.
   - Click No to exit Visual Merge with the working file you had prior to starting Visual Merge. The status of the working file remains Merge.
   - Click Cancel to return to Visual Merge and continue editing.

3. After you exit Visual Merge, if the merged file will change your working file, you are asked whether to overwrite the working file.

   Do one of the following:

   - Click Yes, to have the merged file overwrite the working file. The status of the working file becomes Modified.
   - Click No, to stop the process of saving the merged file as your working file. The status of the working file remains Merge.

4. If you were checking in the file and clicked Yes (in the previous step) to overwrite your working file, you are asked whether to complete the check-in operation.

   Do one of the following:

   - Click Yes to check in the merged file.

     The status of the working file becomes Current.

   - Click No or Cancel to stop the check-in operation.

The status of the working file remains Modified. (If you clicked Cancel, you might need to perform an Update Status operation to see that the status really is Modified.)

Customizing Visual Merge

You can control the appearance of toolbars, line numbers, colors, and so on.

To customize Visual Merge:

1. Select Search > Options from the menu bar. The *Options* dialog appears.
2. Select the Display or the Color tab at the top of the dialog.
   - Select Display to change the appearance of the main window and text in the panes. You can change the tab settings and the font, control the toolbar and status bar, and turn line numbers on or off.
   - Select Color to change the default colors for deleted, changed, inserted, and matching text.
3. Click OK.

Using Visual Merge for Binary Files

The merge operation can be performed on binary files as well as text files. The utility performs the three-way comparison of your working file, the tip revision, and their common ancestor revision. However, when using Visual Merge for binary files, note that:

- You do not use the Visual Merge main window.
- You must decide whether to continue with your working file or the tip revision.

Note It may be more convenient to merge two files' contents using the application in which they were created. For example, Microsoft Word and FrameMaker both offer comparison utilities.

To merge binary files:

1. Select a file with a Merge status.
2. Start a check-in or check-out operation. If the files are binary files, the *StarTeam Cannot Merge Binary Files* dialog appears.

   The Status field indicates whether the source file, target file, or both have changed (in comparison to their common ancestor).
3. Decide which of the two files' contents to use as the merged file by selecting Use Source Contents or Use Target Contents from the Merge Results drop-down list box.
4. Click OK to create the merged file, or click Cancel to avoid creating a merged file at this time.

Note If you are checking the file in, you can check in the merged file or replace the current working file. If you are checking the file out, the merged file replaces your current working file.

Using Visual Diff

You can access Visual Diff:

- As a stand-alone utility with a graphical user interface
- As a stand-alone utility at the command line

- By requesting a comparison of two revisions (or a revision and a working file) in the application.

Visual Diff is a two-way comparison utility that compares text files, binary files and folder contents.

To access Visual Diff as a stand-alone utility with a graphical user interface:

1. Do one of the following:
   - Double-click the Visual Diff icon in the application program group.
   - Select Start > Programs > StarTeam > StarTeam *client_name.x* > Visual Diff.

   The Visual Diff window opens.
2. From the Visual Diff menu bar, do one of the following:
   - Select File > Open Files.

   - Select File > Open Folders.

   The *Open* dialog appears.
3. Select the first file or folder to be compared and then click Open. Another *Open* dialog appears.
4. Select the second file or folder to be compared and then click Open.

Note Visual Diff compares text files or folder contents and displays them in the Visual Diff main window. For binary files, Visual Diff tells you whether the two files are identical or not.

To access Visual Diff as a stand-alone utility from the command line:

- At the command line, enter the following:
  `visdiff first_file_path second_file_path`

Using the Visual Diff Main Window

The Visual Diff main window consists of a menu bar, a tool bar, and two panes. You can display the compared files in the panes, either side-by-side or one on top of the other, or in a merged view in a single-pane window.

For details about the comparison of two text file versions, see "Comparing Files and Viewing Differences" on page 121.

You can also perform an up-and-down comparison of the contents of two folders. This comparison treats the folder contents as text, showing what lines are the same, different, and so on.

Using the Menu Commands

The menu commands differ slightly between the Windows and Cross-Platform clients:

Menu Command	Description and Comments
File > Open Files (*Cntrl+O*)	Opens two files in Visual Diff.
> Open Folders (*Cntrl+E*)	Opens two folders in Visual Diff (Windows client only).
> Close	Closes the current comparison window in Visual Diff.
> Merge Files	Enables you to save the two files as a merged file (Windows client only).
> Reload	Enables you to reload the two files in case either or both of them has changed (Cross-Platform client only).

Menu Command	Description and Comments
> Print Setup	Enables you to set up the printer (Windows client only).
> Print (*Cntrl+P*)	Enables you to print (Windows client only).
> Chart	Displays a bar chart comparing matching, inserted, deleted, and replaced lines.
> Exit	Closes the Visual Diff application.
Edit > Copy (*Cntrl+C*)	Copies the selected text.
>Select All (*Cntrl+A*)	Selects all the text in the source file if the cursor is in the source file (Cross-Platform client only).
> Encoding	Enables you to select an encoding for both files.
> Options	Displays the *Options* dialog. This dialog has different tabs and options in both clients. See "Customizing Visual Diff" on page 124.
> Search Find (*Cntrl+F*)	Searches the file that contains the cursor for the first instance of the specified text after the cursor.
> Find Next (*F3*)	Searches the file that contains the cursor for the next instance of the specified text.
> Next Difference (*F8*)	Searches for the next difference between the two files and displays the first line of the block of text at the top of the pane.
> Previous Difference (*F7*)	Searches for the previous difference between the two files and displays the first line of the block of text at the top of the pane.
> Next Match *(Shift+F8)*	Searches for the next match between the two files and displays the first line of the matching block of text at the top of the pane.
> Next Match *(Shift+F9)*	Searches for the previous difference between the two files and displays the first line of the matching block of text at the top of the pane.
Window > Split Vertically	Splits the pane vertically, showing one file on each side.
> Split Horizontally	Splits the pane horizontally, showing one file above and one below.
> Show Together	Shows the two files together in the same pane (Windows client only).
>Switch Panes; Switch Comparison Order	Switches the panes in which the two files are displayed.
> Differences only	Toggles between showing all the lines and only the lines that differ (Cross-Platform client only). The Windows client performs this action via an option found on the display tab of the *Options* dialog.
> Help > Help Topics	Displays a Windows help file (Windows client only).
> About	Displays the About box (Windows client only).

Changing File Encodings

By default, Visual Diff assumes that the text files being compared have the same encoding as the default code page for the operating system. When the files have different encodings, comparing them is difficult if not impossible until you specify the correct encoding for each file.

To change the encoding for the files being compared:

1. Select Edit > Encoding from the menu bar. The *File Encoding* dialog appears.
2. Select the correct encoding for the first file from the File 1 Encoding list box.

   The first file is to the left or on top, depending on how you are displaying the files. If the two files are displayed jointly in one pane, you may have to split them horizontally or vertically to set the encodings correctly.
3. Select the correct encoding for the second file from the File 2 Encoding list box.
4. Click OK to return to Visual Diff.

Manipulating the Window Splitter Bars with the Mouse

In addition to using the Window menu options and the toolbar buttons to change the display options for the files, you can also manipulate the splitter bar with the mouse.

When in use, the splitter bar divides the Visual Diff main window. When not in use, the horizontal splitter bar is located above the vertical scroll bar and the vertical splitter bar is located to the left of the horizontal scroll bar.

With your mouse, you can do the following:

- Drag the splitter bar that separates the two panes to reproportion those panes.
- Drag the splitter bar that separates two panes to a window edge to remove the split and show the files together in a single pane.
- Drag the vertical splitter bar onto a window that is not split or is split horizontally to split the window vertically.
- Drag the horizontal splitter bar onto a window that is not split or is split vertically to split the window horizontally.
- Double-click a splitter bar that is not in use to activate it.
- Double-click a splitter bar that is in use to deactivate (remove) it.

Comparing Files and Viewing Differences

Whether you launch Visual Diff from within the application or open it as a stand-alone application, Visual Diff automatically compares files or revisions as soon as you identify them. The first file that you select appears on the left and the second file appears on the right.

Note If you access Visual Diff from within the application to compare two revisions of the same file, the older file is on the left and the newer file is on the right.

The file in the left pane (initially, the first file selected or the older revision) is compared to the file in the right pane (initially, the second file you selected). Text in the first file but not in the second is marked as deleted. Text in the second file but not in the first file is marked as added.

If you select the files in the wrong order, you can change the comparison order; that is, you can change which pane displays which file. To change the comparison order, select Window > Switch Comparison Order from the menu bar or click the Switch Comparison Order icon on the toolbar.

If no differences between the two files you select, Visual Diff displays a message stating that the files are the same and does not display them.

After Visual Diff compares and displays the files, the differences are shown in color. The default colors are as follows:

- Black. Matching blocks of text are displayed in black.

- Red. Deleted blocks of text appear in red and struck through. Deleted text appears in the left pane only.
- Green. Inserted blocks of text appear in green. Inserted text appears in the right pane only.
- Blue. Changed blocks of text appear in blue. Similar, but not identical, text is considered changed.

Note If you change the colors used in Visual Diff, Visual Diff checks the background color for you system to ensure that no color conflicts exist. If the background color is the same as one of your colors, Visual Diff uses the system text color instead.

When the compared files are shown in a single pane, matching, deleted, and inserted lines are shown only once in their appropriate color. Changed lines appear twice and are shown in blue. The first line is from the file selected first or the older revision and is struck through. The second line is from the file selected second or the newer revision.

For example:

~~set pvxfid0=T111~~ (older revision)
set pvxfid0=T112 (newer revision)

Searching Text Files

Visual Diff's Find command locates text strings in the active pane. Visual Diff keeps both panes synchronized so that the same string (or its approximate former location) appears in both panes.

You can also search for the next set of matching or non-matching lines or the previous set of matching or non-matching lines by using the Search menu from the Visual Diff menu bar or the toolbar buttons. Visual Diff locates the lines and positions them at the top of each pane.

To find text:

1. Do one of the following:
   - Select Search > Find from the menu bar.
   - Click the Find icon on the toolbar. The *Find* dialog appears.
2. Enter the string to be located in the *Find What* text box.
3. If you want to locate the text exactly as you entered it with regard to case, select the Match Case check box.
4. If necessary, specify a direction for the search.

   Visual Diff remembers the search string until you enter another one or until you close the application. To search in another pane, simply change panes.
5. Click Find Next.
6. To locate the next instance of the text, use Search > Find Next.

To find the next or previous difference, do one of the following:

- Select Search > Next Difference from the menu bar.
- Click the Next Difference icon on the toolbar.
- Select Search > Previous Difference from the menu bar.
- Click the Previous Difference icon on the toolbar.

To find the next or previous match, do one of the following:

- Select Search > Next Match from the menu bar.
- Click the Next Match icon on the toolbar.

- Select Search > Previous Match from the menu bar.
- Click the Previous Match icon on the toolbar.

Merging or Printing Compared Files

When you merge two files or revisions to create a new file, the system uses a series of characters to delineate text rather than using colors. After the files are merged, you must delete extraneous text manually.

Key to Reading Merged File:

#######

The characters above separate changed lines from the rest of the text. Both revisions of the changed lines are shown, separated by **~~~~~~~**.

- - - - - - -

The characters above separate deleted lines from the rest of the text

+++++++

The characters above separate inserted lines from the rest of the text

The following is an example of a merged text file.

```
@ECHO OFF
C:\WINDOWS\NET START
######################### 1 Line(s) Changed ##########
C:\DOS\SMARTDRV.EXE /X
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
C:\DOS\SMARTDRV.EXE 1024 /X
##################################################
PROMPT $p$g
PATH c:\windows;c:\dos;c:\mach32
---------- 1 Line(s) Deleted ---------------------
SET DIRCMD=/oen
--------------------------------------------------
REM Set location for temp files.
######################### 1 Line(s) Changed ##########
SET TEMP=C:\DOS
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
SET TEMP=C:\TEMP
##################################################
++++++++++++++++++++++++ 1 Line(s) Inserted +++++++++
SET TMP=C:\TEMP
++++++++++++++++++++++++++++++++++++++++++++++++++
```

To merge compared files:

1. Open and compare the files.
2. Do one of the following:
   - Select File > Merge Files from the menu bar.
   - Click the Merge icon on the toolbar.

   The *Save Merged File As* dialog appears.
3. Select a folder in which to store the file, then name the file and then click Save.

Visual Diff creates the new file and, if the editing option is set, opens it in Notepad or the text editor of your choice.

Customizing Visual Diff

You can control the appearance of toolbars, line numbers, colors, and so on.

To customize Visual Diff:

1 Do one of the following:

- From the Windows client, select Search > Options from the menu bar.
- From the Cross-Platform client, select Edit > Options from the menu bar.

The *Options* dialog appears.

2 Select one of the tabs at the top of the dialog.

- Select Display to change the main window's appearance and the display of text in the panes. You can:
  - Change the tab settings
  - Change the font (Windows client)
  - Display only the differences (Windows client)
  - Turn line numbers on or off
  - Show or hide the toolbar or status bar
- Select Color to change the default colors for matching, replaced, inserted, and deleted text
- Select Merge to determine when to view a merged file and with what editor (Windows client)
- Select Compare to set search options:
  - Ignore case
  - Ignore whitespace

    With the Windows client, you can ignore or use whitespace by selecting or clearing a check box. When you ignore whitespace, all spaces and tabs on a line, no matter where they appear are excluded from the comparison. This is equivalent to stcmd diff -w at the command line and the VSS style used in the Cross-Platform client version of Visual Diff (explained next).

    With the Cross-Platform client, you can ignore specific types of whitespace.

    The option buttons are:

    - Do not ignore
    - PVCS style. This is equivalent to stcmd diff -bpvcs at the command line. When comparing two lines of text files, Visual Diff ignores leading and trailing whitespace. For example, the following lines are equivalent because there is only one space between "hi" and "mom":

      ```
      "      hi mom   "
      "     hi mom"
      ```

      but the next line is not equivalent:

      ```
      "hi              mom"
      ```

    - VSS style. This is equivalent to stcmd diff -w at the command line and the only Ignore Whitespace option in the Window client. When comparing two lines in text files, Visual Diff ignores all whitespace (tabs and spaces). For example, the following lines would be equivalent:

      ```
      "a = ( b + 2);"
      "a=(b+2);"
      ```

- UNIX diff -b Style. This is equivalent to stcmd diff -b at the command line. When comparing two lines of text files, Visual Diff ignores trailing whitespace and treats all other strings of whitespace as equal in length. For example, the following lines are equivalent:

  ```
  " hi mom   "
  "    hi    mom"
  ```

- (Cross-Platform clients) Ignore end-of-line. Ignoring the EOL enables you to compare files from different platforms without have every line perceived as different. For Windows platforms, the end-of-line marker (EOL) is a carriage return/line feed combination; for UNIX platforms, it is a line feed; for Macintosh files, the end of line varies depending on the application. It may be the traditional Macintosh carriage return or the UNIX line feed.

3 Click OK.

Chapter

8

Managing Files

To place a file under version control, you add it to a project view. After it is under version control, team members can check it out, revise it, and check in a new revision.

The application preserves historical information about each file revision. And because of its linking capabilities, you can link a file revision to other items that affected the file or a particular revision of it.

A number of other operations can be performed on files, such as moving a file or changing its branching behavior. See “Performing Generic Operations” on page 191 for more information.

Using a File Under Version Control

If a file resides in the working folder of an application folder, you can add that file to the application folder. This operation places that file under version control. A copy of the working file becomes the first revision of that file stored in the repository. If the working file is deleted later, the data is not lost because a copy exists in the repository. The application creates a new revision of this file in the repository every time you check the file in.

Every time you check a file revision out, its contents are copied to a working folder. Checking out a revision also ensures that you have the tip or a specific revision to work on. For example, you may need a team member’s most recent changes to a file, or you may have deleted the working file from your hard drive and now need another copy.

Here are some recommendations about using files that are under version control:

- To let other team members know that you intend to make changes to a file, change the lock status to exclusive as part of the check-out procedure.
- As part of the check-in process, you can notify others both that you are finished making your changes to the file and that it is available for them to check out by removing the lock status.
- If you intend to continue making changes to the file but still want to check it in for backup purposes, keep the file locked.
- If two team members change the same text file simultaneously or if one member changes an outdated file, you can use the merge option to combine the changes in

these files so no work is lost. In such cases, the application gives the file a Merge status. For details, see “Merging Files” on page 141.

- To prevent yourself from changing a file that you have not locked, select the Mark Unlocked Working Files Read-only personal option. Then, if you check out a file that you have not locked, the working copy becomes read-only.

The application enables you to label the tip revisions of every item within a view. For example, when the project reaches a particular milestone (such as beta), you might give the view’s items a label, called a view label. Then you can configure the view to return to the way it was at the time the label was applied, check out revisions as a group using that label, create a new view based on the label, or assign the label to a promotion state. See “Using Labels” on page 215 for more information.

The application also has revision or version labels. You can label one or more revisions as you check them in or by applying the label to each of the revisions using the Labels command on the File menu. It is easy to check out those files as a group using the label.

A file revision can have any number of labels. However, no two revisions of the same file in the same view can have the same label.

Using the File List in the Upper Pane

The files that appear in the upper pane depend upon the following factors:

- The application folder you select from the folder hierarchy in the left pane
- The filter you select from the Filter drop-down list box above the upper pane (see “Using File Filters” on page 129 for more information)
- The depth you specify with the All Descendants button or by selecting All Descendants from the File menu
- The sorting, rearranging, and querying you perform on the list (see “Managing Data in the Upper Pane” on page 53 for more information). The application sorts files alphanumerically, regardless of case.
- The files you have excluded from the display by using the folder’s current or inherited exclude list (see “Changing a Folder Name or Description” on page 46 for more information)

After selecting a file from the upper pane, you can perform certain operations on that file by selecting them from the File menu or from the context menu. For example, using File > Edit from the application menu opens that file in Notepad or an alternate editor. In addition, you can:

- Double-click the name of an executable file to run it.
- Double-click the name of a file associated with an application to open it in that application.

Use the tabs in the lower pane to view or edit a file's properties after you select a file in the upper pane. These tabs enable you to:

- Review the properties for this file displayed in the upper pane (click the Detail tab)
- Check its revision history (click History tab)
- Edit revision comments (click the History tab and right-click to display the context menu)
- Compare two file revisions' contents (click the History tab and right-click to display the context menu)
- View the selected file revision's contents in the default editor or in an appropriate application (click the History tab and right-click to display the context menu)
- Review any links created for it (click Link tab)

  (Usually, you link a file to a specific folder, change request, task, topic, or another file.)
- Review, attach, and detach labels (click the Label tab)
- Review the references to the selected file (click the Reference tab)

  A file can be moved from one folder, view, or project to another by dragging the file to its new folder.

  A file can be shared by pressing *Ctrl* and then dragging the file to the second folder. The original and the receiving folders must both use the same server configuration.
- Click a column header to sort the displayed files based on the value in that column. The sort is in ascending numeric, ASCII, alphanumeric, or internal key order, depending on the data. To change the sort order from ascending to descending (or vice versa), click the header a second time. The application displays a triangle in the column header. It points up (ascending order) or down (descending order).
- You can rearrange the fields, show different fields, sort up to four columns, use group bands, and apply queries to the data in the upper pane. You can save the operations you have performed on the data as a query or filter. See "Controlling the Columns" on page 53, "Sorting and Grouping the Data" on page 54, "Using Queries" on page 55, "Creating a Query" on page 56, and "Using Filters" on page 61.
- Press *Shift + F5* to refresh the entire view (all the item lists in all the tabs as well as the folder hierarchy). Press *F5* to refresh only the upper pane. Press *Ctrl + F5* to refresh the upper pane and collapse all open groups simultaneously. Press *F6* to refresh external archives, such as VSS and PVCS.

Using File Filters

Selecting a filter from the Filter drop-down list box enables you to limit the kinds and quantity of files that appear in the upper pane. See "Using Filters" on page 61 for details about creating your own filters.

The application ships with a default set of filters for each component, such as files and change requests. However, depending on your company's initial application release

and the changes made to filters by you and your team members, you may not see all of the default filters listed below.

<All Non-Excluded Files>	Displays all the non-excluded files that exist either in the application folders (whose contents are being displayed) or in their working folders.
<By Status>	Groups the files that have the same status: Current, Missing, Modified, Merge, Out Of Date, Not In View, or Unknown.
Files In View	Displays the files in the working folder that exist in the current project view.
Files Not In View	Displays the files in the working folder that do not exist in the current project view. Unless you add them to the project, their names never appear on the same list as the files that are in your project.
Files To Check In	Displays all the files in the view that need to be checked in. Their statuses are Modified, Merge, or Not In View.
Files To Check Out	Displays all the files in the view that need to be checked out. Their statuses are Out Of Date, Missing, or Merge.

Note The application lists the files that need to be merged when you apply either the Files To Check In or Files To Check Out filter.

Determining the Status of Your Files

The status of your files determine the operations that you can perform on them:

Missing	The file is part of the project view, but it is not in your working folder. If you want to place the file into a working folder on your hard drive, you must check this file out.
Merge	The working file has been altered but is not based on the tip revision (most recent revision) of this file. This usually happens when both you and someone else work on a file at the same time, but the other person checks the file in before you do. When you check this file in, if it is a text file, Visual Merge can reconcile the differences between your working file and the tip revision.
Modified	The working file has been altered and is based on the tip revision of this file. You might want to check this file in so that the tip revision is placed in the application.
Out of Date	The working file is a copy of an old revision of the file. If you need the tip revision, you should check it out.
Not in View	The file is in the working folder but not in the project view. You might want to add this file to the view. If you are no longer using this file, you might want to delete it from the working folder.
Current	The tip revision of this file is in the working folder.
Unknown	The file in the working folder has the same name as a file in the view but the file was not checked out from the repository. You might have copied it from another location. Use Update Status to determine the correct status.

Adding Files to a View

You can add files to an application folder if the files are in the correct working folder.

To add files:

1. Open the correct view.
2. Select a folder from the folder hierarchy.
3. Select the File tab from the upper pane.

4. (Optional) Click the All Descendants button on the toolbar to display all the files that are located in the selected parent folder's child folders.
5. Select the <All Files By Status> or Files Not In View filter from the filter list box on the toolbar.

   If you selected the <All Files By Status> filter, click the button in front of the Not In View group to display the files.

   Not In View status is assigned to each file in a working folder that is not already added to the project (and, therefore, not yet under version control).
6. Do one of the following:
   - Select the files to be added to the project individually.
   - Choose Select > Select All from the File or context menu to select all the files simultaneously.
7. Select Add Files from the File or context menu. The *Add Files* dialog appears.
8. Do one of the following:
   - Enter a generic description for all files in the *File Description* text box.
   - Select the Prompt For Description For Each File check box to write a separate description for each file.
9. From the Lock status group box, select an appropriate option button. Use Exclusive or Non-exclusive to lock the files in your name or select the Unlocked option button to leave the files unlocked at this time.

   Your lock choice lets other team members know whether you are working on the files. An exclusive lock means you intend to change the files.
10. Delete working files removes the associated files from your workstation, while clearing this box to retain these files in your working folder. In this exercise, do not delete the files.
11. If your company enforces a system of process rules, source code and content changes can be made only to meet clearly defined and approved objectives. All files that are added must be linked to specific process items (change requests, requirements, or tasks.

    You may be restricted to a specific type of item or items with specific statuses. If you have the correct access rights, you can select Project > Properties and use the Process Rules tab to see what rules have been selected. Otherwise, this step is optional, and you can select any change request, requirement, or task as a process item.

    To associate the new files with specific process items, select the Link and pin process item check box.

    a. If an active process item has been selected and appears in the Item box, this action accepts it.

    b. If no process item has been selected, or you wish to choose a different process item:

       1. Click the Select button to open the *Select Process Item* dialog.
       2. Select either the List all permitted items or List all permitted items assigned to me option button.
       3. Select the Change Request, Requirement, or Task tab.
       4. Select a specific item as the active process item.
12. Click OK to return to the *Add Files* dialog.
13. If work on the active process item is now complete, select the Mark selected process item as fixed/finished/complete check box.

If you mark a change request as Fixed, the default workflow immediately changes the responsibility to the name of the user who entered the change request.

If you are adding more than one file and not all of the additions are successful, the process item is not marked as fixed, finished, or complete. If you have selected a disabled (read-only) requirement, it cannot be marked complete.

Be aware that marking a change request as Fixed immediately changes the responsibility to the name of the user who entered the change request.

14 Select a label from the Revision label drop-down combo box or create a new revision label by entering its name. Adding or creating a label is useful if you plan to retrieve these files as a group later or if you will need this specific revision of the files.

The existing labels are listed in reverse chronological order based on the time at which they were created.

15 If you are using the Windows client, click OK to add the files.

16 If you are using the Cross-Platform client, you have additional options. To access these options:

- a Click Advanced. This action displays the *Advanced Options* dialog. Although none of the options on this dialog are required, they are useful in specific situations.
- b Select the Perform EOL conversion (CR-LF) check box to control the EOL character that is stored with the files. The default setting for this check box is based on the EOL setting in your Personal Options.
- c Select the appropriate File Encoding from the drop-down list box.
- d Click OK to return to the *Add Files* dialog.
- e Click OK to add the files.

After you add new files to the application, the status of those files changes from Not In View to Current (unless you deleted the working files, in which case, their statuses become Missing).

Depending on the filter you have selected, the names of the selected files disappear from the upper pane or simply change their status to Current. For example, if you used the filter Files Not In View, you can see the files names again by selecting the Files In View filter.

Tip Click the Add Files icon on the toolbar to add files without using the *Add Files* dialog.

Reviewing or Setting the Executable Bit

When you add a file from a UNIX operating system, the state of the executable bit is preserved by the Cross-Platform client. For each file, there is an Executable check box that becomes selected if the executable bit is set and becomes cleared if the bit is not set. Future check-out operations ensure that the executable bit for the checked-out file matches the setting of the Executable check box.

To review or change the state of the executable bit:

1 Select a file from the upper pane.

2 Select File > Properties from the menu bar.

3 From the resulting *File Properties* dialog, review, select, or clear the Executable check box.

Note The Executable bit can be reviewed and set from the Windows client, but it has no effect when the file is checked out with these clients—only when it is checked out with the Cross-Platform client.

Opening Files from the Application

When you double-click a file name, the application does one of the following:

- Runs the file if it is an executable, such as autoexec.bat.
- Displays an error messages if the file is not an executable and there is no associated application.
- Opens the file in an associated application. For example, .doc files will open in Microsoft Word for Windows. However, this only works if an “open” action file association exists for the selected file’s extension.

  A file can be associated with an application in several ways, which include:

  - From Windows Explorer, select View > Options and then using the File Types tab to create or edit a type.
  - From Windows Explorer, select a file then press *Shift+right-click*, then selecting Open With from the context menu.

    If the context menu displays Open with Notepad or Open followed by some other specific application, this is not the same as an Open action file association. Instead, it is an Open With Notepad action, which does not work with the application.

To open a file from the application, do one of the following:

- Double-click the file name in the upper pane.
- Select the file from the upper pane, then select File > Open.

If the file does not open, an association may not have been created for the correct editor.

To set a default editor:

1. Go to Windows Explorer.
2. Select Tools > Folder Options. This action displays the Folder Options dialog.
3. Select the File Types tab.
4. On this tab, highlight the file extension.
5. Click Advanced. This action displays the Edit File Type dialog.
6. Click New to change the setting for files of this type.
7. On the following dialog, enter the required information.
8. Click OK to select the correct editor:

Editing Files from the Application

You can display files from the application in the default editor, Notepad, or an alternate editor. If the file is an executable, such as a .bat file, or has no associated application, you must use the Edit command to edit the file from the application.

To edit a file in the default editor:

1. Select a file from the upper pane.

2 Select File > Edit. The file opens in the default editor, which is Notepad, unless you set an alternate as explained in "Setting File Options" on page 275.

To use an editor other than Notepad on your workstation:

1 Select Tools > Personal Options from the menu bar. The *Personal Options* dialog appears.

2 Select the Files tab.

3 Click the Alternate Applications button.

4 In the resulting *Alternate Applications* dialog:

   a Select the Editor check box.

   b Specify the executable file for your alternate editor.

   c Click OK.

5 Click OK.

Updating File Status

To check the status of a file before checking it in or out, use the Update Status option. This option compares each file in the working folder with the revision you checked out and the tip revision. For example, the upper pane may say that the file is Current, but someone else may have just checked in a copy of the file, so its status really is Out Of Date.

Important

- The Update Status option does not check files in or out. For example, if a file is not in your working folder, updating the status simply lets you know that the file status is set to Missing.
- The Update Status option does not merge files. For example, if a file's status is set to Current before the status update and then set to Merge afterwards, the update status will not merge the files (see "Using Visual Merge" on page 113 for more information).

Unlike refreshes of the upper pane, the Update Status option uses only the MD5 checksum. Refreshes use either the time stamp (and size) or the checksum, depending on the personal file options you have set. See "Setting File Options" on page 275 for more information.

To update file statuses:

1 Select one or more files from the upper pane.

2 Select Update Status from the File or context menu.

If the status currently displayed in the upper pane is incorrect, the application changes it.

Checking In Files

When you check in a working file, the Server creates a new tip revision.

To check in files:

1 On the upper pane, select one or more files.

   If you select files that are Not In View as well as files that are already under version control, the Not In View files will be added to the view as part of the check-in operation.

2 Select File > Check In from the menu bar or context menu. The *Check In* dialog appears.

3. In the Comment text box, describe the changes made to the file. (If you are checking in multiple files, you can also select the Prompt for a comment (check-in reason) for each file check box.)
4. (Optional) To compare the file being checked in with the tip revision of the file in the repository, click the Compare button. If differences exist, this action launches Visual Diff, a utility that shows file changes.
5. To release your lock on the file after check-in, select Unlocked from the My Lock Status group box. If you have locked the file and want to keep it locked, select Keep Current.
6. (Optional) Select Force check-in if you wish to check in a file that is older than the tip revision.
7. If desired, select the Delete working files check box to delete the files from the working folder on your workstation, storing them only in the repository. Clear this check box to retain these files in the working folder, as well as storing them in the repository.
8. (Required if process items are enforced) To link file revisions to a process item, select the Link and pin process item check box. If the use of process items is required, this check box is selected by default.
    a. If an active process item already appears in the Item box, this action accepts it.
    b. If no process item has been selected, or you wish to choose a different process item:
        1. Click the **Select** button to open the *Select Process Item* dialog.
        2. Select one of the following options: List linked items, List all permitted items, or List all permitted items assigned to me.
        3. Select the Change Request, Requirement, or Task tab.
        4. Select a specific item as the active process item.
        5. Click OK to return to the *Check In Files* dialog.
9. (Optional) Select a label from the Revision label drop-down combo box or create a new revision label by entering its name. Existing labels are listed in reverse chronological order, based on the time at which they were created.

    Remember that the same label cannot be used for two revisions of the same file. Therefore, if the label you want to use is currently associated with a previous revision, you must also select the Move the label if it's already assigned to a previous revision check box.
10. If you are using the Windows client:
    a. Click Show Change Requests to review the change requests linked to the files you are checking in.
    b. Click OK to check in the files
11. If you are using the Cross-Platform client, you have additional options. To access these options:
    a. Click Advanced. This action displays the *Advanced Options* dialog.
    b. (Optional) Select the Perform EOL conversion (CR-LF) check box to control the EOL character that is stored with the files. The default setting for this check box is based on the EOL setting in your Personal Options.
    c. In the drop down list box, select the appropriate File encoding.
    d. If desired, click Show Change Requests to review the change requests linked to the files you are checking in.
    e. Click OK to return to the *Check In* dialog.

f Click OK to check in the files.

It is unusual to use both process items and the Show Change Requests button. The differences between the two approaches are:

- One uses an existing link while the other creates a link. For Show Change Requests, no additional linking is done. With process items, linking is done as part of the check-in process. A link is created or updated for every file to the selected change request.
- With the Show Change Requests button, the link's ends may be pinned or floating. With process items, the link's ends are pinned to specific revisions.
- If the change request is marked fixed:
  - With process items, the application creates a link from the process item to the revision of that change request that has the Fixed status.
  - With the Show Change Requests button, the change request is marked as fixed, but the link is not changed in any way. If the link's connection to the change request floats from revision to revision, it is now attached to the new revision of the change request. If it was pinned to a revision of that change request, it mains pinned there.

Tips Click the Check In icon or Check In And Unlock icon on the toolbar to check in files without using the Check In dialog. If process rules are required, the dialog will open anyway.

You can sort and group the change requests by right-clicking the column headers and using the Sort and Group dialog. Clicking an individual header sorts the requests based on the contents of that column.

Checking Out a Revision

When you check out a file, the application copies the requested revision of that file to the appropriate working folder.

If a copy of that file is already in the working folder, it is overwritten unless the working file appears to be more recent than the checked in revision. Then you are asked to confirm the check out.

For example, if you check a file out, edit it, decide you do not want your edits, and check out the file again, you are asked to confirm the check out.

Tip Use the All Descendants button to select files from a hierarchy of folders. (You can also select All Descendants from the File menu.)

You can use Check Out All from the File or context menu to check out all the files in the selected folder and its descendant folders. This is equivalent to selecting All Descendants, selecting all the files in the upper pane, and then selecting Check Out. When you use the Check Out All command, a confirmation dialog appears, regardless of your settings for the personal options for confirmations.

To quickly check out the tip revision:

1 Select one or more files from the upper pane.

2 Do one of the following:

- Click Check Out (black arrow) on the toolbar.
- Click Check Out and Lock (red arrow) on the toolbar.

Either of these actions copies the tip revision to the working folder.

To check out files using the Check Out dialog:

1. Do one of the following:
   - Select one or more files from the upper pane, then select Check Out from the File or context menu.
   - Select Check Out All from the File or context menu.

   The *Check Out* dialog appears.
2. (Optional) Select Force Check-out to overwrite any working files, even if the working files are more recent than those in the repository.
3. Select one of the following options:
   - Current Revision, to check out the tip revision of each selected file.
   - Label, to check out the revision of each of the selected files that has a specific view or revision label. (If the selected file does not have the label, no revision is checked out for that file.) Select the label from the Label drop-down list box.

     The existing view and revision labels are listed in reverse chronological order based on the time for which they were created.The view labels precede the revision labels in the list.
   - Promotion State, to check out the revision of each selected file that has a specific promotion state. Actually, these are the revisions that have the view label currently assigned to this promotion state.
   - As Of, to check out the revision that was the tip revision at the specified date and time.
     1. Click the button between the date and the time to use the calendar.
     2. Specify the date using the next and previous month buttons and the arrow keys.
     3. To specify the time, enter the time or use the spin boxes.
4. Select one of the My Lock Status option buttons or select Keep Current to retain each file's current lock status.

5 To access additional, less frequently used options, click Advanced. The *Advanced Options* dialog appears.

a You can check the files out to a folder other than the designated working folder:

1 Select the Other option button.

2 Enter or browse for a new folder for the checked-out file.

b Select an option button indicating the appropriate EOL character for text files that are being checked out. The options are None, CR-LF, LF, or CR.

For Windows, the EOL character is CR-LF (carriage return/line feed); for UNIX, it is LF (line feed); and for some other operating systems, it is CR (carriage return).

For the Windows client, the default setting is based on the EOL setting for the folder from which the files are being checked out. See "Creating Folders for Other Operating System Files" on page 46.

For the Cross-Platform client, the default setting is based on the EOL setting in your personal options (see page 277 for details).

The default file encoding for these clients is set in personal options, but can be changed here for a specific check out operation.

c Click Close to return to the *Check Out* dialog.

6 Click OK to exit the *Check Out* dialog.

The files are checked out to the specified folder. If they are text files and you have overridden the EOL character, their EOL characters are converted to the specified character or character combination.

7 If you used the Check Out All command, click Yes when the confirmation dialog appears.

Tip The *Check Out* dialog can be resized. For example, if you have long label names and cannot see the entire name in the Label text box, you can drag one side of the dialog to make the dialog larger.

To check out any previous revision of a single file by revision number:

1 Display the history list by doing the following:

a Select the name of the file from the upper pane.

b Select the History tab beneath the lower pane.

2 Select the revision to be checked out.

3 Do one of the following:

- Select Check Out from the *History* context menu.
- Select Check Out To from the *History* context menu.

When you select Check Out To, you can check the file out using a different file name and/or a different path. The *History Check Out* dialog appears.

a To change the file name or path:

1 Select the Other option button.

2 Enter or browse for the new path and/or name for the checked-out file.

b If the file is a text file, select an option button indicating the appropriate EOL character. The options are None, CR-LF, LF, or CR.

c Click OK to exit the *History Check Out* dialog.

The file is checked out to the specified folder with the specified name. If the file is a text file and you have overridden the EOL character, the EOL characters are converted to the specified character or character combination.

Understanding the Effects of Status on Check-ins and Check-outs

During the file check-out process, the application copies a file revision from the repository to a working folder. Checking in a file places a new revision in the repository. In many cases, the status of a file affects the check-in or check-out process.

The following table summarizes the relationship between file status and the check-in and check-out operations for the application. A successful check-in or check-out operation changes the status of your working file to Current (unless you checked out a historical revision or deleted the working file as part of the operation).

Table 8.1 Effects of Status on Check-in and Check-out Operations

Status	Check In	Check Out
Current	No considerations.	No considerations.
Merge	Starts Visual Merge unless you force the check-in. The Merge status means that someone else has checked in this file since your last check out. You do not have their changes in your working file and someone's changes will	Not allowed unless you force the check-out. Your changes will be lost if you check out this file. TIP: You may be able to merge the tip revision and your working file using the application in which this file was created, for example, Word for Windows. If the file is a text file, try a check-in operation.
Missing	Not applicable. If a file has the Missing status, it is not in your working folder so there is nothing to check in.	No considerations. If a file has the Missing status, you are asked if you want to check it out when you open it. You can check it out manually, too.
Modified	No considerations. Unless someone else has the file locked, you can check in the file.	No considerations. Unless someone else has the file locked, you can check in the file.
Not in View	Not applicable. A file with the Not in View status cannot be checked in. You can add it to the project with the Add Files command.	Not applicable. A file with the Not in View status is not in the repository, so there is nothing to check out.
Out of Date	Not allowed unless you force the check-in. Checking in an Out Of Date file means that the tip revision no longer has the changes made to the file since the time your working copy became Out Of Date.	No considerations. Checking out an Out Of Date file makes your working file Current.
Unknown	Not allowed unless you force the check-in. If the file's status is Unknown, the consequences of this action are also unknown. Your working file becomes the tip revision in the repository. Use Update Status with an MD5 checksum to see if the file can be identified. You might want to compare your working file to the tip revision if this is not successful.	Allowed if you merge the file with the tip revision. However, because the very first revision is used as the ancestor file for this merge, it is likely that many, many things appear to have changed or be in conflict. You may prefer to force a check-out (or force a check-in). If the file's status is Unknown, the consequences of this action are also unknown. Your working file is overwritten by the tip revision in the repository. Use Update Status with an MD5 checksum to see if the file can be identified. You might want to compare your working file to the tip revision if this is not successful.

Reviewing the Content of a Revision

You can review the contents of a historical file revision in either the default editor (by default, Notepad) or in the Windows application for which the file type is registered.

To review a revision's contents:

1. Display the history list by doing the following:
   a. Select the name of the file from the upper pane.
   b. Select the *History* tab below the view window's lower pane.
2. Select and right-click the revision in the history list. The History context menu opens.
3. Do one of the following:
   - Click View Revision Content. Clicking this copies the revision to a temporary file and displays the temporary file in the default editor (Notepad or the alternate editor specified in your personal options).
   - Click Open Revision Content. Clicking this copies the revision to a temporary file and displays the temporary file in the appropriate application.

Comparing File Contents

You can make any of the following content comparisons between text files and/or text file revisions. Visual Diff performs the comparison.

To compare two revisions of a file:

1. Display the history list by doing both of the following:
   a. Select the name of the file from the upper pane.
   b. Select the *History* tab below the view window's lower pane.
2. Select two revisions of a file from the history list.
3. Do one of the following:
   - Click the Compare Contents icon on the toolbar.
   - Select Compare Contents from the History context menu.

   Visual Diff, the application differencing program, displays the two revisions, indicating the differences with different colors.
4. Select File > Exit from the Visual Diff menu bar to return to the application.

To compare a revision to the working file:

1. Display the history list by doing both of the following:
   a. Select the name of the file from the upper pane.
   b. Select the History tab below the view window's lower pane.
2. Select one revision of a text file from the history list.
3. Do one of the following:
   - Click the Compare Contents icon on the toolbar.
   - Select Compare Contents from the History context menu.

   Visual Diff displays the two files.
4. Select File > Exit from the Visual Diff menu bar to return to the application.

To compare two working files that are text files:

1. Select a folder from the folder hierarchy.
2. Select the File tab below the upper pane. The upper pane appears in that pane.
3. Select two text files from the upper pane.
4. Do one of the following:
   - Click the Compare Contents icon on the toolbar.
   - Select Compare Contents from the File or context menu.

   Visual Diff displays the two files.
5. Select File > Exit from the Visual Diff menu bar to return to the application.

To compare a text file in the working folder with its tip revision:

1. Select a file from the upper pane.
2. Do one of the following:
   - Click the Compare Contents icon on the toolbar.
   - Select Compare Contents from the File or context menu.

   Visual Diff displays the two files.
3. Select File > Exit from the Visual Diff menu bar to return to the application.

See "Using Visual Merge and Visual Diff" on page 111 for information about searching for differences and other features of the Visual Diff application.

Note — You can compare binary files with Visual Diff as well, but it is much more effective to compare binary files in the application in which they were created. If all you need to know is whether or not they are identical, Visual Diff does this easily and effectively.

Merging Files

In the Windows Client

The application provides file locking and communication tools to help prevent collisions between files that can occur when team members work on files simultaneously. However, it is not always possible to prevent this from happening. For example:

- One team member may begin to make changes before locking a file, only to realize too late that another team member has locked it and made changes as well.
- A team member working off-site might update an out-of-date file without realizing that there is a newer revision stored in the repository.

For text files, the Visual Merge utility helps combine the two files so that no work is lost. This utility merges your working file with the tip revision by comparing the both of them with the revision on which both files are based (that is, the revision that was checked out to create the working file).

Note — For binary files, Visual Merge indicates only if the files are the same or different. You choose which of the two revisions of the file to use as the "merged" file.

Because the merging capability can resolve most of the problems that occur when a text file is worked on by multiple users, you do not always have to exclusively lock these files to prevent such problems.

The application starts Visual Merge as part of the check-in or check-out process if the file's status is Merge. The following procedure explains this process.

To merge files as part of the check-in or check-out process:

1 Check a file in or out (see "Checking In Files" on page 134 or "Checking Out a Revision" on page 136 if you need more information.)

The application displays a message box similar to the following:

Important If the files are binary, you may prefer to stop this operation. You can check out the most currently checked-in file to another location and compare the two files in an appropriate editor. For example, you can compare two .doc files using Microsoft Word for Windows.

2 Click Yes. Visual Merge then performs a three-way comparison.

What happens next depends on:

- The Visual Merge options set for your workstation. For example, the Visual Merge main window may open always—or only when there are conflicts between the two text file revisions (see "Setting File Options" on page 147 for details).
- Whether the file is text or binary

  For text files, the Visual Merge main window can be used to view the merged file created by Visual Merge and edit it.

  For binary files, Visual Merge enables you to choose between your working file and the tip revision.

3 If the Visual Merge window opens:

a See "Using Visual Merge and Visual Diff" on page 111 for information about resolving conflicts in text files and so on.

b Select File > Exit from the Visual Merge menu bar to return to the application.

c The application asks if you have resolved all the conflicts and wish to save the file. It might ask only about saving the file.

d Click Yes to replace your working file with the merged file. The working file's status will change from Merge to Modified.

4 If the file is binary and a dialog similar to the following opens:

a From the Merge Results list box, select Use Source Contents or Use Target Contents.

b Click OK. What happens next depends on whether you check the file in or out.

c If you selected Use Source Contents, the tip revision of this file is copied to your workstation. The working file's status changes from Merge to Current.

d If you selected Use Target Contents, the status of your working file changes from Merge to Modified because the file is considered a merged file.

e If you are checking the file out, the operation now ends. You have the merged file to work with.

f If you are checking the file in, a message box opens.

g Click Yes to check in the merged file.

In the Cross-Platform Client

In the Cross-Platform client, files can be merged only as they are checked out. If you are not using an alternate merge utility, the application displays the two files in a merged format that you can scroll through and edit.

Sections that have conflicts are marked. For example, the following section reveals the first conflict:

```
<<<<<<< C:\StarDraw\Source Code\Starvw.cpp (local)
//MFC C++
=======
//
// This is a part of the Microsoft Foundation Classes C++ library.
>>>>>>> C:\StarDraw\Source Code\Starvw.cpp, Branch Revision: 1.5
```

The lines that start with angle brackets, <<<<<<< and >>>>>>>, identify the two files and separate the conflict from the rest of the file.

Where one file contains “MFC C++”, the other contains “This is a part of the Microsoft foundation Classes C++ library.” You can edit the file to include one, the other, or both lines—whatever is appropriate.

In this example there was only one line in the conflict, but there can be many lines in a conflict and many conflicts.

Depending on the personal options you have set for files, you may see this dialog whenever there are differences of any kind or only when the differences result in conflicts.

When differences do not result in conflicts, they appear in the merged file as they were in the file from which they came. In this example, the lines starting with “This source code is only intended...” and ending “Microsoft Foundation Classes product.” exist only in the local file. This is not specifically marked as a difference in any way.

If you do a comparison prior to the check out, you can review the differences using Visual Diff or an alternate comparison utility.

To merge the working file with an existing revision:

1. Select the file from the upper pane. Its status will be Merge.
2. Select File > Check Out. The *Check Out* dialog appears.
3. Select the tip (current) revision or another revision (by its label, promotion state, or date/time).
4. Click OK.

   The application asks whether or not to merge the revision with your working file.
5. Click Yes.

   The application notifies you whether or not the file contains conflicts. The merge utility appears based on differences or conflicts (depending on your personal option settings for files).
6. If the merge utility appears:
   a. Edit the merged file.
   b. If you are using the merge utility provided by the application:
      1. Select File > Exit.
      2. Click Yes to save the file.

The merged file becomes your working file.

Note Checking out a file from the history pane overwrites the working file with the revision that you selected.

Using Keywords in Files

If keyword expansion is enabled by an administrator as a project property, you can embed keywords within a project text file. The keywords are automatically expanded during file check-out to provide file and revision information within the file. It is recommended that you use only one keyword per line. Keywords expansion should work for all ASCII-based encodings, which includes UTF-8, CP1252, and so on. It does not include the various UTF-16 encodings. StarTeam currently treats UTF-16 encodings as binary and does not attempt to perform either EOL or keyword expansion on them.

The Windows client uses only ASCII characters to expand keywords. Characters outside that range expand to "?". The Cross-Platform client allows you to select a file encoding as you check out a file.

Caution Never use a keyword in a revision comment, because it will be expanded during the keyword expansion process.

The keywords available in the application are:

`$Author$`	User who checked in the revision
`$Date$`	Date and time stamp for the revision
`$DateUTC$`	Same as $Date$ except that a UTC time replaces the local time. UTC times end in a "Z" which makes it readily identifiable. "Z" is the international standard for UTC time. The Z stands for the "zero meridian", which goes through Greenwich in London. It is also commonly used in radio communication, in which it is pronounced "Zulu" (the word for Z in the international radio alphabet).
`$Header$`	Combination of Workfile, Revision, Date, and Author
`$HeaderUTC$`	Same as $Header$ except that a UTC time replaces the local time. UTC times end in a "Z".
`$Id$`	Similar to $HeaderUTC$ except that it is a combination of Workfile, the branch revision number (preceded by a "v" for version, for example, v 1.2.1.0), UTC time, and Author. The branch revision number is in dot notation.
`$Locker$`	User who has the file exclusively locked (if any)
`$Log$ or $Log[x]$`	File change history Log is a special keyword because it expands to a multiline entry. The Log keyword expands to include information for each revision of the file. Revision history includes Revision Number, Date, Author, and Reason for Check In. Use Log to retain entries for each revision within the file. Use $Log[x]$ when you want to limit the number of revisions for which entries are retained. Replace *x* with the number of entries to be retained. For example, $Log[8] saves these entries for the most recent 8 revisions. If you replace *x* with a number less than 1 or with a nonnumeric character, the application ignores *x* and retains all entries (as with Log).
`$LogUTC$ or $LogUTC[x]$`	Same as Log and $Log[x]$ except that a UTC time replaces the local time. UTC times end in a "Z".
`$NoKeywords$`	Turn off keyword expansion for the rest of the file
`$Project$`	Name of the project
`$Revision$`	Revision number (an integer)
`$Folder$`	Name of the folder
`$Workfile$`	Unqualified name of the working file (for example, foo.cpp)

Note for Java Users $Header$ can cause problems if the sequence "\u" appears in the expanded header. For example, suppose you have a file name foo.java stored in the D:\util. The first time you compile it with $Header$, the header is expanded to:

$Header: D:\util\Foo.java, 1, 7/27/99 11:05:48 AM, StarTeam Administrator$

Even though this is contained in a java comment, the java compiler, always looks for \u (for unicode). The second time you compile foo.java, a compiler error occurs.

The following file shows some expanded keywords. Some lines have wrapped onto two lines due to the page size of this manual.

```
$Author: StarTeam Server Administrator$
$Date: 10/20/2003 7:39:00 PM$
$Header: Stardoc.h, 9, 10/20/2003 7:39:00 PM, StarTeam Server Administrator$
$Id: Stardoc.h,v 1.8, 2003-10-21 02:39:00Z, StarTeam Server Administrator$
$Locker:StarTeam Server Administrator$
$Log[5]:
 9    StarDraw  1.8         10/20/2003 7:39:00 PM  StarTeam Server
Administrator
 8    StarDraw  1.7         8/1/1997 5:22:05 PM    Francis Bacon   New Step 8
 7    StarDraw  1.6         7/1/1997 5:15:52 PM    Thomas Edison   New Step 7
 6    StarDraw  1.5         6/1/1997 5:09:21 PM    Albert Einstein New Step 6
 5    StarDraw  1.4         5/1/1997 5:04:58 PM    Nikola Tesla    New Step 5
$
$Project: StarDraw$
$Revision: 9$
$Folder: Source Code$
$Workfile: Stardoc.h$
```

The following `$DateUTC$` keyword is expanded to show the UTC date representation. Note the “Z” following the time.

```
$DateUTC: 2003-01-16 00:50:49Z$
```

You can compare the expanded `$Header$` and `$HeaderUTC$` keywords below.

```
$Header: EnumInfo.cpp, 24, 1/15/2001 4:50:49 PM, TEdison$
$HeaderUTC: EnumInfo.cpp, 24, 2001-01-16 00:50:49Z, TEdison$
```

Note You can display times in the history list in UTC. For details, see “Setting File Options” on page 275.

Modifying File Properties

This section explains how to use the standard property dialog to edit file properties. Depending on how your team has set up the application, you may see a totally different dialog called an alternate property editor (APE).

Every time the properties of a file are modified, a new revision of that file is created. If you modify a property, you should also create a revision comment explaining the modification using the Revision Comment tab.

If you want to rename a file in your project, you should rename it within the application. This retains the properties associated with that file, such as history and links. However, when you roll a view or item back to a configuration prior to the name change, the old name is not restored. This can cause problems. For example a setup script from an earlier build or release might fail because it uses the file’s former name.

If rename a file outside the application (for example, by using Windows Explorer), the application considers the file to be a new file. When you add the file with the new name, it will have no connection to the history, links, or other properties of the original file.

To modify file properties:

1. Select a file from the upper pane.
2. Do one of the following:
   - Select Properties from the File or context menu.
   - Click the Properties icon on the toolbar.

The *File Properties* dialog appears.

3 Select the General tab. It displays:

- The file's name and description (you can modify both of these).

  For example, if you change the file's name from x.cpp to y.cpp, y.cpp becomes this file's name in the project view. The name of the working file, if it exists is changed as well.

- The current status of the working file.

  For example, if the working folder does not contain the file, the file's status is Missing.

- The size of the tip revision in bytes.
- The name of the person who last modified the file and the date and time at which the modification took place.

  The modification could have been a change to the file's properties or a check-in operation.

- The time stamp of the last working file that was checked in.
- The names of those who have the file locked exclusively or non-exclusively.
- For UNIX files, the ability to review or change the state of the executable bit.

4 Select the Working File tab. It displays:

- The path to the working file.
- Whether the file currently resides in the working folder, and, if so, its size (in bytes) and time stamp.

5 Select the Archive tab. The contents of this tab vary, depending upon the file and the client you are using.

a If you use the Cross-Platform client to access a file on a StarTeam 6.0 server or earlier, the fields that appear depend upon whether the file is Native, PVCS, or VSS.

   For Native files, the following fields display:

   - Archive type: Native
   - Compression: Options include None, Maximize Speed, Default (a compromise between speed and compression), or Maximize Compression.
   - Store revisions as deltas: Check box to store text files using the forward delta method. Clear it to store entire file revisions.

   For PVCS or VSS files, the following fields display

   - Archive type: Native, PVCS, or VSS
   - Foreign archive path: Path for PVCS or VSS file

b With the Windows client (any version) or the Cross-Platform client on a StarTeam 2005 or later Server, the fields that appear depend upon whether the file is Native, PVCS, or VSS.

   For Native files, the following fields display:

   - Type: Native
   - Format: Native-I or Native-II
   - Archive File Name: Name of Native File
   - Compression: None or Default

For PVCS or VSS files, the following fields display

- Type: PVCS, or VSS
- Foreign archive path: Path for PVCS or VSS file

6 Select the Custom tab. It displays any customized fields and their values.

If your company purchased the Repository Customization feature, your administrator might have created additional file properties. The *StarTeam Administrator's Guide* provides information about creating custom fields.

a Double-click the name of a custom property. The *Edit Property* dialog appears.

b Select a new value for this property, and click OK.

For integer, text, and real fields, Value is a text box. For enumerated types and user IDs, it is a list box. For dates and times, Value has a date check box and a time check box, each of which is followed by a date or time in the format for your locale.

c Repeat steps a and b for other custom properties.

7 Select the Comment tab. It displays:

- The note or explanation entered for this file revision. You can edit this revision comment (or any other) using the History context menu (see "Reviewing the Content of a Revision" on page 140 for more information).
- A comment text box for the new revision. You should modify this to explain any additional property changes.

  Enter any notes, such as the reason for changing this file's properties, in the Comment For New Revision text box.

8 Click OK.

Setting File Options

A number of personal options are related to files. See "Setting File Options" on page 275.

Chapter

9

Using Requirements

The Requirement component enables you to create requirements within the application and show dependencies among them. For example, if one requirement must be fulfilled before a second requirement can be considered fulfilled, the first can be made a child of the second.

If process rules are enforced and if requirements are being used as process items, your requirements can be used to drive the development process. For more information about process rules, see “Using Process Rules” on page 211

Administrators and other authorized users can export requirements from CaliberRM and then import them into the application. This enables you to integrate advanced requirement management techniques into the development process.

When notification is turned on, those team members responsible for specific requirements will be notified of changes to those requirements.

After the requirements have been finalized, work can be performed to fulfill thee requirements.

A number of other operations can be performed on requirements, such as moving a requirement or changing its branching behavior. See “Performing Generic Operations” on page 191 for more information.

This chapter explains how to use the standard property dialog to create and edit requirements. Depending on how your team has set up the application, you may see a totally different dialog called an alternate property editor (APE).

Displaying Existing Requirements in the Upper Pane

To display existing requirements in the upper pane of the project view window, select the Requirement tab. A Requirement menu also appears on the menu bar. After selecting one or more requirements, you can use the Requirement menu on the menu bar or right-click to display a context menu and perform operations. Double-clicking a single requirement displays its properties.

The requirements in the upper pane have the following characteristics:

- They are attached to the folder selected from the folder hierarchy.
- They match the filter selected from the Filter drop-down list box.

- They match the depth specified by All Descendants. (You can click the button on the toolbar or select All Descendants from the Requirement menu.)

Icons may appear with a requirement to indicate its status and whether you have read the latest revision.

Notes

Press *Shift + F5* to refresh the entire view (all item lists in all tabs as well as the folder hierarchy). Press *F5* to refresh only the upper pane. Press *Ctrl + F5* to refresh the upper pane and collapse all open groups simultaneously.

The default columns for the requirement list are:

Number	Number the application assigned to the requirement when it was first submitted.
Created By	Name of the user who created the requirement
Created Time	Time at which the requirement was created.
Name	Name given to the requirement.
Description	Explanation of the requirement.
Status	Indication of the requirement's progress from submitted to rejected or completed (i.e., the requirement life cycle. The following list shows the sort order for statuses: Submitted Pending Accepted Draft Deferred Review Complete ReadyForCCB (Change Control Board) Rejected Approved
Locked By	Indicates who currently has the requirement exclusively locked.
Priority	Indicates the requirement's importance. The following list shows the sort order for priorities: Unassigned Essential Useful Desirable

---

Click a column header to sort requirements based on their values in that column. Depending on the type of data in a column, the sort is usually in ascending numeric or alphanumeric order.

Here are some tips on working with requirements:

- To change the sort order from ascending to descending (or vice versa), click the header a second time. A triangle indicating the direction of the sort appears on the primary sort column's header.
- You can rearrange the fields, show different fields, sort up to four columns, use group bands, and apply queries to the data in the requirement list. You can save the operations you have performed on the data as a query or filter. See "Controlling the Columns" on page 53, "Sorting and Grouping the Data" on page 54, "Using Queries" on page 55, *"Creating a Query" on page 56*, and "Using Filters" on page 61.
- You can move a requirement from one folder in the folder hierarchy to another by dragging the requirement to its new folder. You can also move requirements that use the same server configuration between views or projects.

- To share a requirement between folders in the folder hierarchy, press *Ctrl* and then drag the requirement to the second folder. You can also share requirements that use the same server configuration between views or projects.

Benefits of Using the Tree Format

With the Tree format, each requirement and its children form an expandable requirement tree so that you can see the relationships between the requirement and its child requirements. The Tree format is the default order for requirements.

You can select Expand All and Collapse All from the *Requirement* menu or context menu to completely expand or collapse all the trees.

Requirements for which you are responsible remain bold (for unread) until they are read. Read and unread requirements can also be identified from the Read and Not Read icons.

If you mark a previously read or updated requirement as unread, it will appear in bold type again.

Benefits of Using the List Format

Using the list format enables you to:

- Sort the requirements
- See the values of specific fields as columns of data. For example, you can see a column of statuses for each requirement, including the icon for each type of status.

Using Requirement Filters

Filtering enables you to limit the kinds and quantity of requirements that appear in the requirement list. Select a filter from the Filter drop-down list box above the requirement list. The information also depends on:

- Your selection from the folder hierarchy in the left pane
- Whether the All Descendants button is selected from the toolbar or Requirement menu

The application ships with a set of default filters for each tab. However, depending on the original release of the application your company started with and the changes made to filters by you and your team members, you may not see all (or any) of the filters listed below.

The default filters are:

<Show All>	Displays all the requirements.
Flagged Items	Lists only requirements that have been flagged
Grouped by Creator	Displays groups of requirements, one group for each user who has created requirements.
Grouped by Status	Displays groups of requirements, one group for each existing status.
I Am Responsible	Displays only the requirements for which you are responsible.

Using the Lower Pane

When you select a requirement from the requirement list, the tabs on the lower pane enable you to:

- Review the properties for this requirement displayed in the upper pane (click the Detail tab)
- Check its revision history (click History tab)
- Edit revision comments (click the History tab and right-click to display the context menu)
- Compare two requirement revisions (click the History tab and right-click to display the context menu)
- Review any links created for it (click the Link tab)

  (Usually, you link a requirement to a specific folder or another item.)
- Review, attach, and detach labels (click the Label tab)
- Review the references to the selected file (click the Reference tab)

Creating a Requirement or Child Requirement

Creating a hierarchy of requirements enables you to organize a project and shorten development time by preventing costly, time-consuming misunderstandings.

To create a requirement:

1. Do one of the following:
   - To create a new requirement that is not the child of an existing requirement, from the Requirement menu or context menu, click New.
   - To create a new requirement that is the child of an existing requirement, from the Requirement menu or context menu, click New Child Requirement.

   The *New Requirement* dialog appears.
2. Use the Requirement tab to:
   a. Provide a name for this requirement.
   b. Select an owner (person ultimately responsible for the fulfillment of the requirement from the owner drop-down list box.
   c. (Optional) Provide an external source or reference for this requirement in the External Reference text box.

      (If you are importing requirements from CaliberRM, this field displays its identification for this requirement.)
   d. Enter the initial description of this requirement. The description is usually revised over time to eliminate ambiguities.
3. Use the Responsibilities tab to list the team members responsible for this requirement. These people will be notified of changes to the requirement if notification is enabled.
   a. Click Add to display the Select Responsible Users dialog.
   b. Double-click the name of each person to be added to the list.

      As you double-click the name, it moves from the Users list box to the Responsible list box.
   c. When all the responsible users have been added to the Responsible list box, click OK.
4. Later reviewers will use the Ambiguity Review tab to locate ambiguities in the initial description and revise that description. You can ignore this tab for the time being.
5. Use the Estimate tab to indicate the “best case” and “worst case” times for fulfilling this requirement. The entries are usually in staff days.

a. Enter the number of units (usually days) estimated for the fulfillment of this requirement in the Expected text box.

b. Enter the number of units (usually days) estimated for the “worst case” fulfillment of this requirement in the High Effort text box.

c. Enter the number of units (usually days) estimated for the “best case” fulfillment of this requirement in the Low Effort text box.

d. Add any appropriate notes in the Notes text box.

6. Use the Custom tab to provide values for requirement properties that your team leader or company may have created. The *StarTeam Administator’s Guide* provides information about creating custom fields.

   a. Double-click the name of a custom property. The *Edit Property* dialog appears.

   b. Select a new value for this property, and click OK.

      - For integer, text, and real fields, Value is a text box.
      - For enumerated types and user IDs, Value is a list box.
      - For dates and times, Value has a date check box and a time check box, each of which is followed by a date or time in the format for your locale.

   c. Repeat steps a and b for other custom properties.

7. (Optional) Use the Attachments tab to add additional files to this requirement.

   a. Click Add if you want to attach a file. For example, you want to show what a user interface should look like.

      The *Open* dialog appears.

   b. Select a file to be attached from the *Open* dialog.

   c. Click Open.

   d. Repeat the previous steps for additional attachments. You can attach a maximum of 64 files to an item.

8. Use the Comment tab to explain why the requirement is being created.

   Other team members should add comments as they revise the requirement to explain their changes. Enter the reasons in the Comment For New Revision text box.

9. Click OK to save the requirement.

Exporting Requirements from CaliberRM

You can export requirements from CaliberRM. You use a custom CaliberRM Document Factory report. For more information, see your CaliberRM documentation.

Importing Requirements from CaliberRM

To import requirements from CaliberRM into StarTeam, use the CaliberRM StarTeam Publisher tool. This tool lets you automatically create and updates requirements in StarTeam that correspond to requirements in CaliberRM.

You download this tool from http://starteam.borland.com/starteamintegrations. See the *CaliberRM StarTeam Publisher User Guide* that is installed with the CaliberRM StarTeam Publisher tool for more information.

Displaying a CaliberRM Requirement in StarTeam

If you use Enterprise Advantage, the sample alternate property editors (APEs) for requirements include a button that will start the CaliberRMViewer.

When CaliberRMViewer is used, it displays a dialog that enables you to edit the requirement in CaliberRM.

Chapter 10

Tracking Change Requests

A change request is a request to change something within the scope of a project. For example, you might suggest a product enhancement or request a fix for an error or problem. To use the change request tracking system effectively, you need to understand the model on which it is based.

This chapter explains how to use the standard property dialog to create and edit change requests. Depending on how your team has set up the application, you may see a totally different dialog called an alternate property editor (APE).

Important

Even if you use the standard property dialog for change requests, your company or team leader may have a set of guidelines for using change requests that is different from the model presented below. In that case, the model serves only as an example.

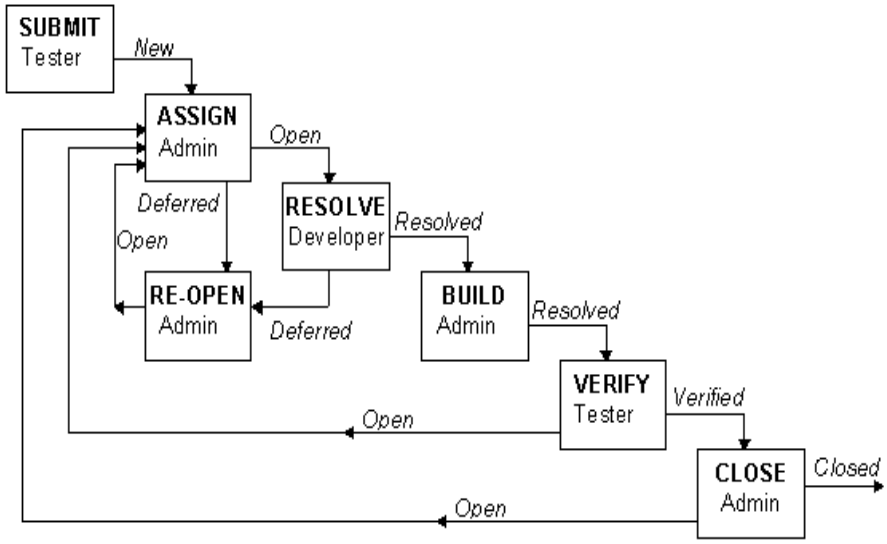
A number of other operations can be performed on change requests, such as moving a change request or changing its branching behavior. See “Performing Generic Operations” on page 191 for more information.

Important Even if you use the standard property dialog for change requests, your company or team leader may have a set of guidelines for using change requests that is different from the model presented below. In that case, the model serves only as an example.

Change Request Tracking System Model

The following diagram and steps show the change request tracking process. The boxes represent the steps taken from the time that the change request is submitted until the time it is closed. Each box indicates an action and the team member most

likely to be responsible for performing this action. The arrows show the status of the change request at the time of each step.



The change request tracking system consists of the following:

1 A team member creates a new change request that does either of the following:

- Summarizes a problem with your company's product and lists the steps to be taken to reproduce the problem.
- Suggests an enhancement to the product.

This change request has a status of New.

2 Another person, such as an administrator or team leader, decides whether to fix the problem or add the suggested enhancement to the product.

This person can:

- Set the change request's status to Open and assign a team member to resolve it.
- Set the change request's status to Deferred because it is worthwhile but will not be done at this time.
- Set the change request's status to Is Duplicate because this is not the first time it has been submitted.

  If desired, a link can be created between a change request and the original submission so that the change request can be tracked along with the original submission.
- Set the change request's status to As Designed because the product is supposed to work this way, meaning there is no defect.

Change requests with a status of Open go to step 3.

3 The person assigned to resolving the change request changes the status of the change request to In Progress. Later on, after this person is finished examining the change request, he or she changes the status to one of the following:

- Fixed
- Documented
- Cannot Reproduce.

4 Next, a team member (usually a tester) verifies the change request. For example, a test case may be developed to determine if the problem is really fixed, documented or not reproducible and changes the status to one of the following verified statuses:

- Verified As Designed
- Verified Cannot Reproduce
- Verified Documented
- Verified Fixed
- Verified Is Duplicate

5 Finally, yet another team member changes the status to Closed. This person may then perform activities related to closing the change request, such as retesting the change request before closing it or adding it to a report to be included in the next release of the product.

In most of the above steps, the change request can be reopened and reprocessed.

Built-in Workflow

The application has a built-in workflow for change requests that automatically sets many of the values associated with change requests. This built-in workflow determines these settings based on the change request's Status field.

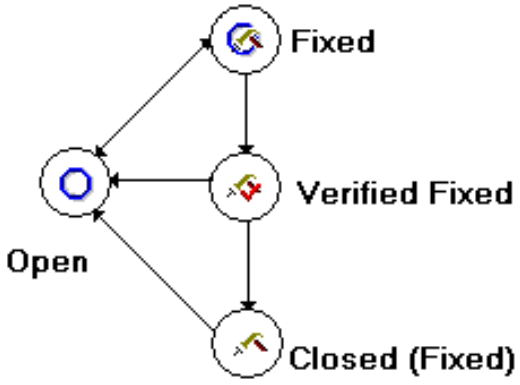
You cannot add additional settings to the Status field. However, you can rename them to better suit preferences set by your organization. For example, your organization may prefer to change the name of the Status "New" to "New Change Request".

When you alter the status of a change request, the built-in workflow automatically selects the appropriate properties associated with the change in status. For more information and a table that summarizes automatic workflow changes, see the section "Summary" on page 166.

After New, Open, or In Progress is selected, six new statuses are created in the Status drop-down list box. These statuses, which are associated with the status you selected, are:

- Deferred
- Cannot Reproduce
- As Designed
- Fixed
- Documented
- Is Duplicate

The next diagram shows the life cycle for a change request with an initial status of Open. The status was then set to Fixed. After this setting, the built-in workflow added an additional status field of Verified Fixed. Finally, the change request was closed, meaning its status was set to Closed (Fixed).



The diagram also shows that a change request can be reopened at any stage in its life cycle because the arrows leading from each of the three fixed statuses can lead back to the Open status at any time. An open change request is represented by an icon in the form of a circle.

The diagram also shows icons for each of the three fixed statuses:

- A closed toolbox, representing a status of Fixed.
- A closed toolbox with a small green check mark in the lower left corner, representing a status of Verified Fixed.
- A closed toolbox with an 'x' in the lower left corner, representing a status of Closed (Fixed).

Displaying Existing Change Requests in the Upper Pane

To display existing change requests in the upper pane of the project view window, select the Change Request tab. A Change Request menu also appears on the menu bar. After selecting one or more change requests, you can use the Change Request menu on the menu bar or right-click to display a context menu. Double-clicking a single change request displays its properties.

Icons may appear with a change request. The icons indicate the change request type, status, and severity level.

The change requests in the upper pane all:

- Are attached to the folder selected from the folder hierarchy.
- Match the filter selected from the Filter drop-down list box.
- Match the depth specified by All Descendants. (You can click the button on the toolbar or select All Descendants from the Change Request menu.)

Note Press *Shift + F5* to refresh the entire view (all the item lists in all the tabs as well as the folder hierarchy). Press *F5* to refresh only the upper pane. Press *Ctrl + F5* to refresh the upper pane and collapse all open groups simultaneously.

Default columns for the change request list are:

- **CR Number**

  Number the application assigned to the change request when it was first submitted.

- **Synopsis**

  Short explanation of the change request

- **Type**

  Whether the change request is for a defect or a suggestion

- **Status**

  Indication of the change request's progress from open to resolved to verified to closed. This is referred to as a change request's life cycle. Statuses include: New, Open, In Progress, Resolved, Verified, and Closed.

- **Cannot Reproduce**

  Indicates that the defect cannot be reproduced.

- **As Designed**
- **Fixed**
- **Documented**
- **Is Duplicate**

- **Severity**

  Seriousness of the change request. Severity can be High, Medium, or Low.

- **Responsibility**

  Person responsible for the change request

- **Addressed In**

  Build in which the change request is resolved.

- **Addressed By**

  Name of user who resolved the change request. A change request has been resolved when its status becomes Cannot Reproduce, As Designed, Fixed, Documented, or Is Duplicate.

- **Last Build Tested**

  Build in which the change request occurs.

- **Work Around**

  Solution to change request other than a fix.

- **Modified Time**

  Time the change request was last modified.

- **Priority**

  Indication that the change request is a priority or not a priority.

- **Description**

  Complete explanation of the defect or suggestion, including the steps to be taken to reproduce the problem.

- **Test Command**

  Command that can be used to test the change request's solution.

- **Fix**

  Solution to the problem addressed by the change request.

- **Entered On**

  The date and time the change request was submitted.

- **Entered By**

  Person who submitted the change request.

Click a column header to sort change requests based on their values in that column. Depending on the type of data in a column, the sort is usually in ascending numeric or alphanumeric order. Other sort orders include:

- Severity: Its ascending order goes low to medium to high.
- Status: Its sorted based on its life cycle and has an ascending order from new to closed.
- Priority: Its ascending order goes from No to Yes.

To change the sort order from ascending to descending (or vice versa), click the header a second time. A triangle indicating the direction of the sort appears on the primary sort column's header.

You can rearrange the fields, show different fields, sort up to four columns, use group bands, and apply queries to the data in the change request list. You can save the operations you have performed on the data as a query or filter. See "Controlling the

Columns" on page 53, "Sorting and Grouping the Data" on page 54, "Using Queries" on page 55, "Creating a Query" on page 56, and "Using Filters" on page 61.

Using Change Request Filters

Filtering enables you to limit the kinds and quantity of change requests that appear in the change request list. Select a filter from the Filter drop-down list box above the change request list. The information also depends on:

- Your selection from the folder hierarchy in the left pane
- Whether the All Descendants button is selected from the toolbar or Change Request menu

The application ships with a set of default filters for each tab. However, depending on the original release of the application your company started with and the changes made to filters by you and your team members, you may not see all (or any) of the filters listed below.

The default filters are:

<Show All>	Displays all the change requests.
By Status and Responsibility	Groups change requests based on their statuses and the users who are currently responsible for processing the requests.
Not a Priority	Displays only the change requests that are not a priority.
Priority	Displays only the change requests that are a priority.
Show Unread Changes	Displays only the change requests that you have not read (or not read since they were modified)
Status = Closed	Displays only the change requests that are closed.
Status = Deferred	Displays only the change requests that are postponed.
Status = Open	Displays only the change requests that are open and in progress.
Status = Resolved	Displays all the change requests that have one of the following statuses: As Designed, Cannot Reproduce, Documented, Fixed, or Is Duplicate.
Status = Verified	Displays all the change requests that have one of the following statuses: Verified As Designed, Verified Cannot Reproduce, Verified Documented, Verified Fixed, or Verified Is Duplicate.
Type = Defect	Displays only the change requests that have the type Defect
Type = Suggestion	Displays only the change requests that have the type Suggestion.

Using Existing Change Requests

Change requests that appear in bold type signify that you are responsible for them and that they have not yet been read. Change requests in regular type are those that you have read or for which you are not responsible.

If you mark a previously read or updated change request as unread, it will appear in bold type again.

You can move a change request from one folder in the folder hierarchy to another by dragging the change request to its new folder. You can also move change requests that use the same server configuration between views or projects.

To share a change request between folders in the folder hierarchy, press *Ctrl* and then drag the change request to the second folder. You can also share change requests that use the same server configuration between views or projects.

Using the Lower Pane

When you select a change request from the change request list, the lower pane's tabs enable you to:

- Review the properties for this change request displayed in the upper pane (click the Detail tab)
- Check its revision history (click History tab)
- Edit revision comments (click the History tab and right-click to display the context menu)
- Compare two change request revisions (click the History tab and right-click to display the context menu)
- Review any links created for it (click Link tab)

  (Usually, you link a change request to a specific folder, file, task, topic, or another change request.)
- Review, attach, and detach labels (click the Label tab)
- Review the references to the selected file (click the Reference tab)

Submitting a Change Request

Change requests are submitted so their resolutions can be tracked. Your team may already have a set of guidelines for filling out fields in a change request. For example, you might have specific instructions on what kinds of information can be typed in the change request's Component and Category fields.

To create a change request:

1. Select a folder from the folder hierarchy.
2. Select the Change Request tab on the upper pane. The change request list appears in the upper pane.
3. Do one of the following:
   - Select New from the Change Request or context menu.
   - Click the New icon on the toolbar.

   The *New Change Request* dialog appears.
4. Accept the default status New, or select another status from the Status drop-down list box.
5. Indicate the severity of the change request by selecting High, Medium, or Low from the Severity drop-down list box.

   The team leader usually identifies the criteria for change requests with a status of high, medium and low.
6. If the change request needs immediate attention, select Yes from the Priority list box.
7. Specify the type of change request by selecting Defect or Suggestion from the Type list box.
8. Select the platform which the defect or suggestion applies to from the Platform drop-down list box.
9. (Optional) The External Reference text box can be used when importing change requests from StarTeam 2.1 or Version 2.0 projects. Alternatively, you might use it to indicate which customer's call to support resulted in the entry of this defect.

The External Reference text box can contain a maximum of 64 characters.

10 (Optional) Indicate the build that the change request occurs in by selecting a build from the Last Build Tested drop-down list box. The default is the name of the most recent build.

11 (Optional) The Component text box is intended for differentiating change requests. For example, differentiating whether or not the change request applies to the client or server.

The Component text box can contain a maximum of 64 characters.

12 (Optional) The Category text box is intended to group change requests by additional criteria such as dialog name, menu, etc.

The Category text box can contain a maximum of 64 characters.

13 Enter a summary of the change request in the Synopsis text box.

The application accepts a maximum of 20K characters in this text box.

14 Select the name of the team member responsible for correcting the change request from the Responsibility drop-down list box.

15 Select the Description tab.

a Enter a full description of the change request, including the steps to reproduce it, in the Description text box.

The application accepts a maximum of 20K characters in this text box.

b (Optional) Enter or browse for the path to a test for the change request in the Test Command text box.

16 (Optional) Select the Solution tab. Enter a temporary solution in the Work Around text box. The Work Around text box is usually completed by the user who fixes the code.

The application accepts a maximum of 20K characters in this text box.

17 Select the Custom tab.

If your company purchased the Repository Customization feature, your team leader may have created additional change request properties. The *StarTeam Administrator's Guide* provides information about creating custom fields.

a Double-click the name of a custom property.

b In the resulting *Edit Property* dialog, select a new value for this property, and click OK.

- For integer, text, and real fields, Value is a text box.
- For enumerated types and user IDs, it is a list box.
- For dates and times, Value has a date check box and a time check box, each of which is followed by a date or time in the format for your locale.

c Repeat steps a and b for other custom properties.

18 (Optional) Select the Attachments tab.

a Click Add to attach a file to the change request. For example, you might attach a screen capture of a problem.

The *Open* dialog displays.

b Select a file to be attached from the *Open* dialog.

c Click Open.

d Repeat the previous steps for additional attachments. You can attach a maximum of 64 files to an item.

To remove an attachment by selecting it and click Remove.

19 (Optional) Select the Comment tab.

20 Enter any notes, such as the reason for changing this change request's properties, in the Comment For New Revision text box.

21 Click OK to save the new change request.

Assigning Change Requests

To assign change requests, you must select the Change Requests tab, the correct folder from the folder hierarchy in the left pane, and if necessary, the All Descendants toolbar button. You may have created a filter or query to find all the change requests that have New as their status or you may sort the Status column in the upper pane.

To assign change requests:

1 Do one of the following:

- Double-click the change request.
- Select the change request, then click the Properties icon on the toolbar.
- Select the change request, then select Properties from the Change Request or context menu.

The *Change Request Properties* dialog appears.

2 Review the settings and decide on an appropriate status. You can make the status Open, Is Duplicate, As Designed, or Deferred.

3 If you make the status Open, you can also change the Responsibility property to the person best qualified to fix or enhance the product as described in this change request.

4 Do one of the following:

- Click Apply, then click the Next or Previous button to review another change request.
- Click OK to close this dialog.

When the status of a change request becomes Is Duplicate or As Designed: the application automatically changes the responsibility to the person who submitted the change request. The assumption is that the person who submitted the change request will want to know about, verify, or perhaps challenge this change in status.

Note The person who processes new change requests should also periodically process deferred change requests—for example, when a new release of the product is planned.

Resolving Open Change Requests

To locate change requests that you need to resolve, select the Change Requests tab, the correct folder, and if necessary, the All Descendants toolbar button. Like most users, you will probably create a filter or query that locates all change requests for which you are currently responsible (see "Managing Data in the Upper Pane" on page 53 for more information).

Depending on the processes used by your team or organization, you may be required to link open change requests to the file or files that will need to be changed to for fix the change request. Also, before you start working on the change request, you may be required to change its status to In Progress.

As you check in a file or group of files, you can indicate what change requests are fixed by the files being checked in. This saves the time it would have taken to change the status of each change request separately using another defect tracking system or even using the change request list in the upper pane of the view window. For details, see "Checking In Files" on page 134.

To resolve a change request:

1. Double-click the change request. The *Change Request Properties* dialog appears.
2. Change the status to one of the following resolved statuses:
   - Fixed
   - Documented
   - Cannot Reproduce.

   You might also use Is Duplicate or As Designed—if these facts were not noticed by the person who gave the change request the status Open.
3. If the status is Fixed or Documented, select the Solution tab, and enter the appropriate information in the Work Around and/or Fix text boxes.

   The application can accept a maximum of 20K characters in each of these text boxes.

   Often a change request suggests one or more fixes for the problem and none of them are the fix that you implement. To avoid confusion, you must describe your fix exactly. Testers and writers rely heavily on this information.
4. Click OK to close this dialog.

The application makes the following changes automatically when the status of a change request becomes resolved:

- The responsibility for the change request is changed to the person who submitted the change request. The assumption is that the person who submitted the change request will want to know about, verify, or perhaps challenge this change in status.
- If the status changes to Fixed or Documented, the setting for the Addressed In Build property automatically becomes "Next Build." Then when the next build label (a special type of view label) is created, the application changes "Next Build" to the name of the build label. The assumption is that the person verifying that this change request has been implemented should test the correct build of the product.

Tip Because the application makes these automatic changes immediately, you may want to change the Responsibility or Addressed In Build setting before you click Apply or OK. Doing this enables you to bypass the automatic workflow and route the change request as your team requires. For example, all resolved change requests in your system may need to become the responsibility of someone in quality assurance, regardless of who submitted them.

Verifying Resolved Change Requests

To verify a change request, you test it in the correct build of the product. If a test command is associated with the change request, you can run that command from the change request. In many cases, the Test Command property will be empty. You may have to devise a new test for the change request or run one or more existing tests.

To mark the change request as verified, select the Change Requests tab, the correct folder, and if necessary, the All Descendants toolbar button. Like most users, you will probably create a filter or query that locates all the change requests for which you are currently responsible. Then you can click the Status column to sort these by status—if you didn't already build that into your filter.

If you determine that the change request is not really resolved, you can reopen it.

To run the test associated with a change request:

1. Double-click a change request from the change request list.
2. Select the Description tab.
3. If the *Test Command* text box has an entry, click Run.

To mark a change request as verified or to reopen it:

1. Double-click a change request.

   The *Change Request Properties* dialog appears.
2. Change the status to Open or Verified.

   The application has the following verified statuses:

   - Verified As Designed
   - Verified Cannot Reproduce
   - Verified Documented
   - Verified Fixed
   - Verified Is Duplicate
3. If the status is set to Open, add the word “Reopen” followed by the date to the change request’s synopsis. Otherwise, the team member who resolved this change request may think that he or she forgot to mark it resolved and, without investigating it further, mark it resolved a second time.
4. Click OK to close this dialog.

Automatic changes made to a change request when the status becomes reopened:

- If the status is Open, the application changes the responsibility to the person who resolved the change request.

  The assumption is that the person who resolved the change request the first time should be the person to continue working on it.
- The setting for the Addressed In Build property becomes blank.

  The assumption is that the change request is not resolved and, therefore, has not been addressed in any build.

Closing Verified Change Requests

To close change requests, you must select the Change Requests tab, the correct folder, and if necessary, the All Descendants toolbar button. You may have created a filter or query to find all the change requests that have Verified as their status, or you may sort the Status column in the upper pane.

To close a change request:

1. Double-click a change request.

   The *Change Request Properties* dialog appears.
2. Change the status to Closed.

   The application has the following closed statuses:

   - Closed (As Designed)
   - Closed (Cannot Reproduce)
   - Closed (Deferred)
   - Closed (Documented)

- Closed (Fixed)
- Closed (Is Duplicate)

3 Do one of the following:

- Click Apply, then click the Next or Previous button to review another change request.
- Click OK to close this dialog.

If necessary, you can reopen a change request at this point. See “Verifying Resolved Change Requests” on page 164 for details about what information to add to the synopsis and the automatic workflow changes if you reopen a change request.

Summary

The following table summarizes the steps used in processing change requests as explained in this chapter. It includes the automatic workflow changes the application makes to change requests based on their statuses.

Table 10.1 Change Request Process Management

Step	Description
Submit	Anyone (usually a tester) can submit a change request. Process: Select the Change Request tab. Then select New from the Change Request or context menu. A change request has the following default properties (which you can change if necessary). Status: New Severity: Low Priority: Not prioritized Type: Defect Platform: All Last Build Tested: Current build label Entered by: Person currently logged on to the application Many other fields are initially blank. Some team leaders prefer to have all change requests submitted at the root folder. They use drag-and-drop to move the change requests to the appropriate child folders.
Assign*	Process: The team leader finds all new change requests and does one of the following: ■ Opens the change request and assigns it to a developer, help writer, or other appropriate team member. ■ Defers the change request until a later date, perhaps the next release of the product. ■ Specifies that the change request is “As Designed” and not to be fixed. If the change request status is Open, no automatic changes occur. If the change request status is Deferred or As Designed, then Addressed in Build is disabled and the responsibility is assigned to the user who created the change request.

Table 10.1 Change Request Process Management (continued)

Step	Description
Resolve*	Process: Users find the Open or In Progress change requests assigned to them, and do one of the following for each request: Resolve the problem in the system and update the change request's properties. (The statuses that indicate that a change request has been resolved are Cannot Reproduce, As Designed, Fixed, Documented, or Is Duplicate.) Defer the change request until a later date, perhaps the next release of the product. Your team leader may prefer that you do not defer change requests. If the change request status is one of the possible resolution statuses, then Addressed in Build becomes Next Build for Fixed and Documented statuses. It becomes disabled for other statuses. By default, the responsibility is assigned to the person who submitted the change request, who is expected to verify the resolution. If the change request status is Deferred, then Addressed in Build is disabled and the responsibility is assigned, by default, to the user who created the change request.
Build	Who builds the project? The project view may have a formal or informal build process. However, at some point, all the files, etc. currently in the view receive that build label. It is usually applied to the source code files, etc. that were compiled (and may need to be changed) rather than to the executable files that result from the build. Effect on change requests: For any resolved change request that has Next Build as the setting for its Addressed In Build property, "Next Build" is replaced with the next build label that is created. **Note:** If a new build label is based on a past configuration (rather than the current configuration), it has no effect on the Addressed In Build property. If a change request has not branched in its current location, "Next Build" may be replaced with a build label from another view. For example, suppose you create a branching child view or share a folder from one view to another. Suppose that "Next Build" is the value of some change request's Addressed In Build property and that change request has not branched. When a build label is created in the source view, "Next Build" is replaced with the name of that build label, regardless of the location.

Table 10.1 Change Request Process Management (continued)

Step	Description
Verify*	The person who submitted the change request (usually a tester) verifies a resolution. Process: Install the build in which the resolution is to be verified and determine whether the change request has been resolved correctly. Do one of the following: Verify the change request, marking it as Verified Cannot Reproduce, Verified As Designed, Verified Fixed, Verified Documented, or Verified Is Duplicate. Re-open the change request and update the setting for Last Build Tested. If the change request status is Verified, no automatic changes occur. If the change request status is Open, Addressed in Build is blank. If the change request has changed from resolved to Open, the user who changed the status to Fixed or Documented becomes responsible.
Close*	Usually the team leader closes the change request. Process: The team leader does one of the following: Reviews and closes the verified change request. Re-opens the change request. If the change request status is Closed, then no automatic changes occur. If the change request status is Open, then Addressed in Build is blank. If the change request has changed from resolved to Open, the user who changed the status to Fixed or Documented becomes responsible. **Then**: Addressed in Build: blank Responsibility: for a change from verified to Open, the user who changed the status to Fixed or Documented becomes responsible.

*Changes in status can result in automatic changes to other properties.

Using Reports and Charts

You can create a number of charts and reports using the change request report and change request chart features. This section contains two samples and explains how to save these setting for future use as a filter. For more information on using reports and charts, see “Reporting and Exporting Data” on page 249 and “Using Charts” on page 259.

The following examples instruct you to use the *Show Fields*, *Sort and Group*, and *Queries* commands. For more information about these commands, see “Managing Data in the Upper Pane.”

To create a report or chart that shows all new change requests and their authors:

1. Select the root folder from the folder hierarchy in the left pane.
2. Select the Change Request tab.
3. Select All Descendants from the toolbar or the *Change Request* menu.
4. Right-click the column headers and select Show Fields from the context menu.
5. In the resulting *Show Fields* dialog:
   a. Make sure the CR Number, Entered By, and any other appropriate fields are displayed in the change request list.
   b. Click OK.
6. Right-click the column headers and select Sort and Group from the context menu.
   a. From the *First By* group box, select Status from the drop-down list box.

b Select the Group By check box.

c From the first of the *Then By* group boxes, select Entered By from the drop-down list box.

d Select the Group By check box.

e Click OK.

7 Select the New grouping.

8 Do one of the following:

- Select Reports from the Change Request or context menu. See "Reporting and Exporting Data" on page 249 for more information.
- Select Charts > Distribution from the Change Request or context menu. See "Using Charts" on page 259 for more information.

To create a report or chart about the defects fixed during a certain time period:

1 Select the root folder from the folder hierarchy in the left pane.

2 Select the Change Request tab.

3 Select All Descendants from the toolbar or the *Change Request* menu.

4 Right-click the column headers and select Show Fields from the context menu.

5 In the resulting *Show Fields* dialog:

a Make sure the CR Number, Status, Modified Time, Modified By, and any other appropriate fields are displayed in the change request list.

b Click OK.

6 Right-click the column headers and select Queries from the context menu.

7 In the resulting *Queries* dialog:

a Create a query to find all change requests with a Modified Time greater than or equal to the earliest date of interest.

b If the end date is not the current date, AND a condition requiring the Modified Time to be less than or equal to the end date.

c Click OK.

8 Right-click the column headers and select Sort and Group from the context menu.

a From the *First By* group box, select Status from the drop-down list box.

b Select the Group By check box.

c Click OK.

9 Select the *Fixed* group.

10 Do one of the following:

- Select Reports from the Change Request or context menu. See "Reporting and Exporting Data" on page 249 for more information.
- Select Charts > Time Series from the Change Request or context menu. See "Using Charts" on page 259 for more information.

After you have sorted, grouped, selected columns, applied queries, and so on, you can save the arrangement of change request data that appears in the upper pane as a filter.

To create a filter for the current change request arrangement:

1 Right-click the column headers and select Save Current Settings from the context menu. The *Save Current Settings* dialog appears.

2 Enter a filter name in the *Filter Name* text box.

3. Select or clear the *Public* check box depending on whether this filter is to be used by all or only on your workstation.
4. Click OK. The filter name will appear from now on in the Filter drop-down list box.

Chapter:

11

Managing Tasks

The Task component enables local and remote users to report their efforts on the tasks that they have been assigned. This component can be used as a standalone or interoperate with Microsoft Project. In each case, the Task component is used differently. This chapter covers how to use the Task component as a standalone. For information about interoperating with Microsoft Project, see the *StarTeam Microsoft Project Integration User's Guide.*

Depending on your client or server license, the Task component may not be available to you.

As a standalone, the Task component is very useful for managing a project. With tasks, you can outline who should do what and when. You can estimate the time it will take to complete work on a task, record how many hours a task actually takes, and compare the times. Incorporating the project schedule into the same application as the version control system and the defect tracking system enables you to link tasks to files or change requests and perform several other useful operations.

A number of other operations can be performed on tasks such as moving or sharing them. See "Performing Generic Operations" on page 191 for more information.

Important This chapter explains how to use the standard property dialog to create and edit tasks. Depending on how your team has set up the application, you may see a totally different dialog called an alternate property editor (APE).

Using Tasks in the Upper Pane

When you select the Task tab, the tasks appear in the upper pane in either a tree or list format. In addition, the tasks displayed depend on the following:

- What application folder you select from the folder hierarchy in the left pane
- The filter you select from the Filter drop-down list box above the upper pane (see "Using Task Filters" on page 173 for more information)
- The depth you specify with the All Descendants button or by selecting All Descendants from the Task menu
- The sorting, rearranging, and querying you perform on the list (see "Managing Data in the Upper Pane" on page 53 for more information). Tasks can be sorted only when they are in list mode.

The status bar below the lower pane displays the total number of tasks and the number of tasks selected in either the tree or list view.

When you select a task or subtask in either the list or tree format, the lower pane's tabs enable you to:

- Review the properties that would be displayed in the upper pane if the list format was selected (the Detail tab)
- Check its revision history (the History tab)
- Edit revision comments (the History tab and right-click to display the context menu)
- Compare two task or subtask revisions (the History tab and right-click to display the context menu)
- Review any links created for it (the Link tab)

Important Tasks are not automatically linked to their subtasks, but you can link them. Usually, you would link a task to a specific folder, file, topic, or change request.

- Review, attach, and detach labels (click the Label tab)
- Review the references to the selected task or subtask (click the Reference tab)

Tips

- To move a task tree or one of its branches from one folder in the folder hierarchy to another (or from one view to another), drag the task or branch to its new folder. If you move a task, it becomes a task in its new location.
- Use *Ctrl + drag-and-drop* to share tasks and subtasks.
- Double-click a task to display its properties.
- Use the Task menu or right-click in the task tree pane to display a context menu.
- Press *Shift + F5* to refresh the entire view, and *F5* to only refresh only the upper pane. Press *Ctrl + F5* to refresh the upper pane and collapse all open groups simultaneously.
- Use the List Display button to view the tasks and subtasks in the list format.
- Use the Tree Display button to view the tasks and subtasks in the tree format.

Benefits of Using the Tree Format

The default order for the tasks and work assignments is their tree order. Each task and its set of subtasks form an expandable task tree.

Using the tree format enables you to see the relationships between the task and subtasks. For example, a subtask must be completed before completing a task. You can also mark and entire branch as read or unread.

Selecting Expand All or Collapse All from the *Task* menu or context menu completely expands or collapses all the trees.

Tasks and subtasks for which you are responsible remain bold (for unread) until they are read. Read and unread tasks can also be identified by their icons. Icons indicate the following:

- A StarTeam task and subtask that has been read - represented by an icon with a lined pad on a clipboard with an opened envelope decorator in the lower left corner.
- An Unread StarTeam task and subtask- represented by an icon with a lined pad on a clipboard with a closed envelope decorator in the lower left corner.
- A Microsoft Project task and subtask that has been read - represented by an icon with a checkerboard pad on a clipboard with an opened envelope decorator in the lower left corner.
- An Unread Microsoft Project task and subtask - represented by an icon with a checkerboard pad on a clipboard with a closed envelope decorator in the lower left corner.

Benefits of Using the List Format

You can click List Display to display the tasks as a list, which allows them to be sorted.

Using the list format enables you to:

- See the values of specific fields as columns of data. For example, you can see a column of statuses for each task, including the icon for each type of status.
- Sort the tasks or subtasks.

You can click a column header to sort the rows based on the value in the column. Click a second time to reverse the order. For example, clicking the Created Time column to see the most recently entered tasks.

With the list format, you can also:

- Rearrange fields, show different fields, sort up to four columns, use group bands, and apply queries to the data in the task list and save these operations for later use, such as displaying certain data as a query or filter (see “Controlling the Columns” on page 53, “Sorting and Grouping the Data” on page 54, “Using Queries” on page 55, “Creating a Query” on page 56, and “Using Filters” on page 61).
- Query and filter the task tree.

Using Task Filters

Selecting a filter from the *Filter* drop-down list box makes it easy to locate tasks or subtasks that match that filter.

The default filter is:

- <Show All> (identifies all the tasks)

You can customize your task filters. For example, using the *“Responsibility Equals <user name>*” filter identifies only tasks that someone is responsible for using, while the “*Percent Complete Less Than 100*” *filter* identifies unfinished tasks.

For more information about creating filters, see “Managing Data in the Upper Pane” on page 53

Creating a Task or Subtask

You can create an individual task or a summary task that has a set of subtasks. This requires some planning. After a task has work added to it, no subtasks can be added to it.

You can create one task tree in an application folder or several, each of which may have children, grandchildren, etc. For example, suppose one task tree is for the first service pack that will be shipped with your product. The task tree might look as follows:

- Service Pack 1
  - New features
    - New feature 1
    - Coded
    - Tested
    - Documented
    - New feature 2
    - Coded
    - Tested
    - Documented
  - Major fixes
    - Fix 1
    - Coded
    - Tested
    - Documented
    - Fix 2
    - Coded
    - Tested
    - Documented
  - Preparations for web site downloading
  - Announcements to press and current customers

The service pack's release would be a milestone. It is certainly a task with subtasks, so no work is added to it. Similarly, the tasks New features, Major fixes, etc. will have no work added to them because each must have subtasks. The work will be added to the subtasks that have no additional subtasks, such as Coded, Tested, Documented.

Whether or not work can be assigned to a task, you still assign a team member to be responsible for the task's completion. For tasks to which work can be assigned, you estimate how long it should take and provide other initial information.

To create a task or subtask:

1. Select a folder from the folder hierarchy.
2. Select the Task tab.

   The task menu appears on the menu bar and tasks appear in the upper pane.
3. Do one of the following:
   - Select New from the *Task* or context menu. This creates a new task, which is a root of a task tree.

- Click the New icon on the toolbar. This creates a new task, which is a root of a task tree.
- Select New Subtask from the Task or context menu. This creates a subtask for the selected task.

  If the selected task already has work records added to it, the New Subtask command is disabled.

The *New Task* dialog appears with the Task tab selected.

4 Enter the name of the task in the *Name* text box.

The Name text box can contain up to 255 characters.

5 Select the name of the team member responsible for the completion of this task from the *Responsibility* drop-down list box.

Important The Responsibility field is assigned to the person responsible for the completion of a task. However, other people can be assigned as additional resources. For example, if the completion of this task relies on the completion of subtasks that are assigned to other team members, you might assign those team members as resources by selecting the Resources tab.

6 Select the Milestone check box to indicate that the task should be treated as a milestone task.

You can display the Milestone column in the task list and sort for the tasks that are designated as milestones.

7 Select the current status of the task from the *Status* drop-down list box. Possible statuses include:

- . Pending

  Default for new tasks indicating that the task is awaiting the completion or start of a predecessor task.
- Ready to Start

  This task may be started because all predecessor tasks have been completed.
- In Progress

  The status indicates that work has been entered for this task.
- Finished

  This task is finished according to the team members working on the task.
- Closed

  The task is completed to the satisfaction of the team leader or task reviewer.
- Hold

  This task can have work performed for it based on the state of its predecessor tasks, but it has been placed on hold due to other scheduling issues.

8 Select the importance of the task from *Priority* drop-down list box. These priorities are identical to those used in Microsoft Project.

- Highest
- Very High
- Higher
- High
- Medium (default value)
- Low
- Lower

- Very Low
- Lowest
- Do Not Level

Important The Do Not Level option is a Microsoft Project-specific term that you should ignore.

9 Enter the number of hours the task will take to complete in the Duration text box. This field is disabled if the task contains subtasks, because the duration of a parent task is dependent upon the duration of subtasks.

10 Enter the percentage of work that has been completed on a task in the Percent Complete text box.

The Percent Complete text box can hold values ranging from 0 to 100 and defaults to 0 for new tasks.

11 (Optional) Select the Needs Attention check box to notify team leaders or task reviewers that this task requires attention. Enter the information about why this task needs attention in the text box.

Team leaders can add the Needs Attention column to the task list and sort for these items.

12 This procedure continues with the next section "Assigning Task Resources."

Assigning Task Resources

As you create a task or subtask, you can assign additional team members (resources) to assist in the completion of the task. Use the Resources tab to view the list of team members available for this task assignment.

To assign team members to tasks:

1 Select the Resources tab.

2 Click Add. The *Select Task Resources* dialog appears.

3 Select the team members you want assign to the task to from the Users list, then click Add.

The selected team members appear in the Assigned Resources list.

4 Click OK to return to the Resources tab.

The list of team members you added as resources appear in the *Task Resources Assignment* list.

5 Do any of the following:

- If the Task Resources Assignment list is correct, click Apply.
- To remove a resource from the Task Resource Assignment list, select the name and click Remove.
- If you need to add another resource, click Add and follow steps 3 through 4.

6 This procedure continues with the next section "Estimating the Time Required to Complete a Task."

Estimating the Time Required to Complete a Task

The Time tab enables you to estimate the amount of time needed to complete the task. You can set estimated and actual times for the task. The application does not provide a rollup mechanism for estimate, actual, and variance values.

To set the estimated time to complete a task:

1. Select the Time tab.
2. Do one of the following:
   - Use the *Start* drop-down calendar to enter a start date for the task.
   - Enter the date (mm/dd/yy) in the *Start* text box.
3. Do one of the following:
   - Use the *Finish* drop-down calendar to enter the finish date for the task.
   - Enter the finish date (mm/dd/yy) in the *Finish* text box.
4. Enter the estimated work (hours) required to complete this task in the Work field.

   For example, if you are calculating that this task with take three days and that a day is eight hours, type 24 in the *Work* text box.

   The rest of this dialog is disabled. Microsoft Project would use these fields, but you cannot.

Adding Work to a Task or Subtask

If you work on a task, you can add a work record to indicate what you did and how long it took. For example, suppose your task is to invite the media and you have been given a list of 25 reporters to call. If you work on this task for one hour one day and for three hours the next, you would enter two work records, one for each day. You might also make additional changes to the task. For example, if you've had a problem reaching one reporter, you might want to make a note of that using the Notes tab or bring this problem to the attention of others using the Needs Attention check box on the Task tab.

To add work to a task or subtask:

1. Do one of the following:
   - Double-click the task. From the resulting *Task properties* dialog, select the Work tab, then click Add.
   - Select the task. Select Task > Add Work from the menu bar.
   - Select the task. Click the Add Work icon on the toolbar.

   The *Work Record* dialog appears and displays your name in the User Name drop-down list box.
2. Enter or select the date for the work record from the drop-down calendar.
3. Enter the number of hours you worked in the *Work* text box.
4. Enter the number of hours it will take to complete the work for the task in the *Remaining Work* text box.
5. Enter an explanation about the progress that has been made on task in the Comments text box.

To edit a work record:

1. Select the Work tab.
2. Select the work record from the *Work Records* list.
3. Click Edit. The *Work Record* dialog appears.
4. Edit any field, then click OK.

To delete a work record:

1. Select the Work tab.

2 Select the work record from the Work Records list.

3 Click Delete. The message “Delete Work Record?” appears.

4 Click Yes.

Adding Notes to a Task

Use the Notes tab to provide additional information about the task.

To add notes to the task:

1 From the *Task* properties dialog, select the Notes tab.

2 Enter additional information or notes pertaining to the task in the Notes text box.

Setting Custom Property Fields for Tasks

Depending on your client or server license, repository customization may not be available to you. If your administrator has created additional task properties, you may be required to fill them in for each task. The *StarTeam Administrator's Guide* provides information about creating custom fields.

To set custom property fields:

1 From the *Task* properties dialog, select the Custom tab.

2 Double-click the name of a custom property.

The *Edit Property* dialog appears.

3 Select a new value for this property, and click OK.

- For integer, text, and real fields, Value is a text box.
- For enumerated types and user IDs, Value is a list box.
- For dates and times, Value has a date check box and a time check box, each of which is followed by a date or time in the format for your locale.

4 Repeat steps 2 and 3 for other custom properties.

Including Attachments

Use the Attachments tab to include screen shots and documentation with your task.

To add an attachment to a task:

1 Select the Attachments tab.

2 Click Add. The *Open* dialog appears.

3 Select the file, then click Open. The selected item appears in the *Attachments* list.

4 Repeat steps 2 and 3 for additional files.

You can attach a maximum of 64 files to an item.

To remove an attachment from a task:

1 Select the Attachments tab.

2 Select the file from the *Attachments* list.

3 Click Remove. The attachment is removed from the task.

To save the attachment with a new name:

1 Select the Attachments tab.

2 Select the file from the *Attachments* list.

3 Click Save As. The *Save As* dialog appears.

4 Enter the new name for the file, then click Save.

Including Comments for Task Revisions

After you modify one or more properties in the $Task$ properties dialog and click OK, the application creates a new revision of the task. You should add a revision comment or note explaining why you made the revision prior to clicking OK.

To provide a revision comment:

1 Select the Comment tab.

2 Enter a revision comment.

Modifying Task Properties

You can view and edit task properties at any time provided you have security privileges to do so.

To modify a task's properties:

1 Do one of the following:

- Double-click the task in the task list or task tree.
- Select a task then click the Properties icon on the toolbar.
- Select a task then select Properties from the Task or context menu.

The *Task* properties dialog appears.

2 Select any tab (Task, Time, Work, Notes, Custom, Attachments, or Revision Comment) and begin editing.

Chapter 12

Managing Topics

Topics can raise general questions about the project or start very specific discussions about issues, such as feature implementation. While the responses can lead to resolution of these issues, the historical value of these conversations to the project can be even more significant. Future team members can:

- Reassess decisions more capably
- Avoid retrying solutions that were previously found faulty
- Understand why a particular solution to a problem became necessary and, therefore, not replace that solution with one that doesn't meet all the necessary criteria

Team members can use topics to:

- Search topics and responses for specific words or phrases
- Sort the topics and responses
- Filter the topics and responses
- See the relationships between the topic and its responses
- Move and share topics (from the tree format)

The upper pane consists of topics and a series of responses to each topic. A series of topic trees are eventually formed, each of which consists of a root topic and its responses. The topic tree resembles a conversation that may go on among several people. In the application, this is called a threaded conversation because a topic and its responses are threaded together, starting with the root topic.

By reading each response in a thread, one after the other, and the responses to those responses, you can see how the discussion has evolved.

A number of other operations can be performed on topics or responses such as moving or sharing them. See "Performing Generic Operations" on page 191 for more information.

Important This chapter explains how to use the standard property dialog to create and edit topics and responses. Depending on how your team has set up the application, you may see a totally different topic dialog, called an alternate property editor (APE).

Using Topics in the Upper Pane

When the Topic tab is selected, the upper pane displays topics in either a tree or list format. In addition, the topics displayed depend on all of the following:

- The folder that you select from the folder hierarchy in the left pane
- The filter you select from the Filter drop-down list box above the upper pane (see "Using Topic Filters" on page 183 for more information)
- The depth you specify with the All Descendants button or by selecting All Descendants from the Topic menu
- The sorting, rearranging, and querying you perform on the list (see "Managing Data in the Upper Pane" on page 53 for more information). Topics can be sorted only when they are in list mode.

The status bar below the lower pane displays the total number of topics and the number of topics selected in either the tree or list view.

When you select a topic or response in either the list or tree format, the lower pane's tabs enable you to:

- Review the properties that would be displayed in the upper pane if the list format was selected (the Detail tab)
- Check its revision history (the History tab)
- Edit revision comments (the History tab and right-click to display the context menu)
- Compare two topic or response revisions (the History tab and right-click to display context menu)
- Review any links created for it (the Link tab)
- Review, attach, and detach labels (click the Label tab)
- Review the references to the selected file (click the Reference tab)

Here's how to work with topics:

- To move a topic tree or one of its branches from one folder in the folder hierarchy to another (or from one view to another), drag the topic or branch to its new folder. If you move a topic, it becomes a topic in its new location.
- Use *Ctrl* plus drag-and-drop to share topics and responses.
- Double-click a topic to display its properties.
- Use the Topic menu or right-click in the topic tree pane to display the context menu.
- Press *Shift* + *F5* to refresh the entire view or *F5* to refresh only the task list. Press *Ctrl* + *F5* to refresh the upper pane and collapse all open groups simultaneously.
- Use the List Display button to view the topics and responses in the list format.
- Use the Tree Display button to view the topics and responses in the tree format.

Benefits of Using the Tree Format

The default order for the topics and responses is their tree order. Each topic and its responses form an expandable topic tree:

Using the tree format enables you to see the relationships between the topic and its responses. For example, a response may comment on the topic itself or on another response.

You can select Expand All and Collapse All from the *Topic* menu or context menu completely expands or collapses all the trees.

Icons indicate:

- Topics that have been read.
- Unread topics or topics for which some response has not been read.
- Responses
- Unread responses or responses that have unread responses somewhere in their branches. The red flags lead you to unread items.

Benefits of Using the List Format

Using the list format enables you to:

- See the values of specific fields as columns of data. For example, you can see a column of statuses for each topic, including an icon that indicates whether the topic is active or inactive.
- Sort the topics and responses

Tip You can click a column header to sort the rows based on the value in the column. Click a second time to reverse the order. For example, you can click the Created Time column to see the most recently entered topics and responses.

- You can rearrange the fields, show different fields, sort up to four columns, use group bands, and apply queries to the data in the topic list. You can save the operations you have performed on the data as a query or filter. See "Controlling the Columns" on page 53, "Sorting and Grouping the Data" on page 54, "Using Queries" on page 55, "Creating a Query" on page 56, and "Using Filters" on page 61. You can also query and filter the topic and response trees.

Using Topic Filters

Selecting a filter from the Filter drop-down list box makes it easy to locate topics and responses that match that filter. The application ships with a set of default filters for each tab. However, depending on the release of the application that your company originally licensed and the changes your team made to the filters, you may not see any or all of the filters listed below.

The default filters are:

<I Am Recipient>	Identifies all the topics that name you as a recipient.
By Creator	
Show Active	Identifies all the topics and responses that have Active status.
Show All	Identifies all the topics and responses.

You can customize your topic filters. Some suggestions are:

Created By = <user name>	Identifies only the topics that someone (you) submitted.
Hot Topics = <Priority = High>	Identifies only the topics and responses that were assigned a high priority.

You can customize your topic filters. Some suggestions are: *Created By* = <*user name*>, to identify only the topics that someone (you) submitted and *Hot Topics* = <*Priority* = *High*>, to identify only those topics and responses that have been assigned a high priority.

Creating a Topic

To start a threaded conversation, you must first create a topic.

To create a topic:

1. Select a folder from the folder hierarchy.
2. Select the Topic tab. The Topic menu appears on the menu bar and the topics appear in the upper pane.
3. Do one of the following to create a topic:
   - Select New from the *Topic* or context menu.
   - Click the New icon on the toolbar.

   The *New Topic* dialog appears with the Topic tab selected.
4. Enter a title in the *Title* text box.
5. Enter your communication in the *Content* text box.
6. (Optional) Select the Options tab if you want to notify only specific team members about this topic, assign a priority, or indicate a status.
   a. To notify specific team members about a topic, click Add. The *Select Topic Recipients* dialog appears.

      All team members can read topics, but the notification process can be limited to a few essential people.

   b. Select the team members from the list, then click Add.

   c. Click OK to return to the *New Topic* dialog.

   d. To assign a priority to the topic, select either low, normal or high from the *Priority* drop-down list box.

   e. To indicate a topic status, select either Active or Closed from the drop-down list box. The topic status default is active.
7. (Optional) Select the Custom tab.

   If your company purchased the Repository Customization feature, your administrator might have created additional topic properties. The *StarTeam Administrator's Guide* provides information about creating custom fields.

   a. Double-click the name of a custom property. The *Edit Property* dialog appears.

   b. Select a new value for this property, and click OK.
      - For integer, text, and real fields, Value is a text box.
      - For enumerated types and user IDs, Value is a list box.
      - For dates and times, Value has a date check box and a time check box, each of which is followed by a date or time in the format for your locale.

   c. Repeat steps a through b for other custom properties.
8. (Optional) Select the Attachments tab to attach a file, such as a graphic, to your topic.
   a. Click Add. The *Open* dialog appears.

   b. Select a file to be attached from the *Open* dialog.

c Click Open.

d Repeat the previous steps for additional attachments.

You can attach a maximum of 64 files to an item.

Note You can remove an attachment by selecting the attachment and clicking Remove.

1 (Optional) Select the Comment tab to add additional notes or a comment in the *Comment For New Revision* text box.

2 Click OK.

When the upper pane is in tree format, an icon, the topic's title, your user name, and the time stamp appear in the topic tree pane.

When the upper pane is in list format, the default columns in the topic list display the same information. The topic list has one additional column, Description, which displays the first few words in the topic's text.

Replying to a Topic or Response

To participate in a threaded conversation, you respond to the topic or one or more of its responses.

To respond to a topic or response:

1 Select the topic or response from the topic list or topic tree pane.

2 Click Respond on the Topic or context menu. The *New Topic* dialog appears.

3 Enter a title in the *Title* text box.

4 Enter your communication in the *Content* text box.

5 Follow steps 6 through 10 in "Creating a Topic" on page 184.

Modifying Topic and Response Properties

You can modify the following topic and response properties:

- Title and text of the topic or response
- Priority (such as low, normal, or high)
- Status (such as active and inactive)
- Recipients (list of who received the topic or response)
- Attachments (additional files included with the topic or response)
- The values of any custom fields
- Add additional revision comments or notes

Important After you modify a property and click Apply or OK, the application creates a new revision of the topic or response. You should also add a revision comment or note explaining why you made the revision.

If your only change to a topic is to delete recipients, you cannot delete yourself as a recipient—unless you delete all recipients. When no recipients are specified, everyone, including you, is notified about this topic, so you can delete yourself. When recipients are specified, the application does not allow you to remove yourself as a recipient of the topic.

To modify topic or response properties:

1 Select the folder to which the topic or response is attached from the folder hierarchy.

2 Select the *Topic* tab from the upper pane.

3 Do one of the following:

- Double-click the topic or response.
- Select the topic or response, then click the Properties icon on the toolbar.
- Select the topic or response, then select Properties from the *Topic* menu, context menu or *History* context menu.

The *Topic Properties* dialog appears.

4 Change the settings for the properties.

Chapter 13

Using the Audit Log

The audit log is a chronological record kept by the application. It accumulates data about actions performed on folders and items. Audit log entries cannot be moved, shared, modified, or branched.

A number of other operations can be performed on audit log entries, such as searching for an audit entry by its properties or sending one from the application as e-mail. See “Performing Generic Operations” on page 191 for more information.

If you select the Audit tab (on upper pane) and the audit log is empty, your administrator has disabled the audit log from the server.

If the audit log is enabled, it lists the entries that:

- Are attached to the folder selected from the folder hierarchy
- Match the depth specified by All Descendants (You can click the button on the toolbar or select All Descendants from the Audit menu.)
- Match the filter selected from the Filter drop-down list box

Each entry in the audit log contains several fields, each of which has its own column header. The columns usually displayed are:

User	Person performing the action.
Created Time	Date and time of the action.
Class Name	Type of item: file, folder, change request, topic or task.
Event	Name of the action, such as Created, Added, Deleted, Modified, Branched, Moved From, Shared, Locked, Unlocked, Moved to, or Lock broken.
View	Name of the view in which the item is located.
Project	Name of the project in which the view is located.

Note A filter or query can locate all the entries for a particular item. For example, to view all the entries associated with change request number 1030, create a query that searches for every entry for every item with the number 1030.

Click a column header to sort the displayed entries based on the value in that column. The sort is in ascending order, numeric, alphanumeric, or by key depending on the data. To change the sort order from ascending to descending (or vice versa), click the header a second time. The primary sort column’s header displays a directional arrow.

Use the All Descendants button to select audit entries from a hierarchy of folders. (You can also select All Descendants from the Audit menu.) Select Audit > Select > All from the Audit menu or context menu to select all the entries simultaneously.

Select an audit entry. The *Details* tab on the lower pane displays data about the entry in the lower pane. Because an audit entry has no history and cannot be linked, the *History* and *Link* tabs are not available.

After selecting one or more entries, use the Audit menu or right-click to display the context menu.

You can rearrange the fields, show different fields, sort up to four columns, use group bands, and apply queries to the data in the audit log. You can save the operations you have performed on the data as a query or filter. See "Controlling the Columns" on page 53, "Sorting and Grouping the Data" on page 54, "Using Queries" on page 55, "Creating a Query" on page 56, and "Using Filters" on page 61.

Press *Shift + F5* to refresh the entire view or *F5* to refresh only the audit list. Press *Ctrl + F5* to refresh the upper pane and collapse all open groups simultaneously.

When you select an audit entry from the audit log in the upper pane, the lower pane's tabs enable you to review the properties for this audit entry displayed in the upper pane (click the Detail tab)

A number of other operations can be performed on audit log entries, such as searching for an audit entry by its properties or sending one from the application as e-mail. See "Performing Generic Operations" on page 191 for more information.

Using Audit Log Filters

Filtering enables you to limit the kinds and quantity of entries to be displayed in the audit log. Select a filter from the Filter drop-down list box above the audit log. The items displayed are affected by your selection from the folder hierarchy and by whether you have clicked the All Descendants button.

The default filters are:

Show All	Displays all the audit entries.
By Class and Event	Displays audit entries sorted by the value in the Class Name 1 field (folder, file, change request, topic, task, label, or link) and by the value in the Event (added, modified, deleted, moved from, moved to, shared, created, locked, unlocked, comment edited, label created, label attached, promotion model modified, etc.)

Tip You can create and customize filters. See "Using Filters" on page 61 for more information.

Some suggestions for creating audit filters:

- To display only the entries that were modified, use the Event field.
- To display only the entries about actions that were performed on change requests, use the Class Name field.
- To display only the entries created from the specified date to present, use the Created Time field.
- To display only the entries about actions performed by a specific user, use the User field.

Audit Log Events

Events are actions performed on an owner. For example, a file can be checked in or removed from version control. Such events are recorded in the audit log. Most items can be:

- Added
- Branched
- Comment Edited
- Created
- Deleted
- Locked
- Lock Broken
- Modified
- Moved From
- Moved To
- Shared
- Unlocked

In addition, files can be:

- Converted
- Edited
- Item Overwritten (as foreign archive files become native files)
- Vault

Labels can be:

- Created
- Modified
- Deleted
- Frozen
- Unfrozen
- Attached
- Moved
- Detached

  I

Promotion models can be:

- Modified

Promotion states can be:

- Modified

Chapter

14

Performing Generic Operations

A number of operations performed on folders and component items (files, change requests, requirements, tasks, and topics) are similar in their effect and are therefore considered generic operations. The following table offers a brief overview of the generic operations that can and cannot be performed on folders and components. Many of these operations are explained in more depth later in this chapter. However, operations specific to that component appear in the chapter for that component.

Table 14.1 Generic Operations for Folders and Items

Operation	Folder	File	Change Request	Requirement	Task	Topic	Audit
Moving Drag an item to a new location	Yes Moving a folder moves its contents, child folders, and their contents.	Yes	Yes		Yes, except when using Microsoft Project Integration	Yes	No
Creating shortcuts to items	No	Yes	Yes		Yes	Yes	No
Copying items to a third-party application via a URL (Cross-Platform client only) See “Opening and Copying StarTeam URLs” on page 195.	Yes	Yes	Yes	Yes	Yes	Yes	No
Sharing *Ctrl+drag* item to a new location.	Yes	Yes	Yes	Yes	Yes	Yes	No
Branching behavior Branching occurs at the view level, but branching behavior (i.e., whether or not an item branches when it changes can be controlled at the item level	Yes Branching a folder does not branch its contents or children	Yes	Yes	No	No	No	No

Table 14.1 Generic Operations for Folders and Items (continued)

Operation	Folder	File	Change Request	Requirement	Task	Topic	Audit
Configuring to or freezing at a point in the past	Yes	Yes	Yes	Yes	Yes	Yes	No
Locking	No	Yes	Yes	Yes	Yes	Yes	No
Comparing properties of two items of the same type	No	Yes	Yes	Yes	Yes	Yes	No
Comparing properties of two revisions	Yes	Yes	Yes	Yes	Yes	Yes	No
Review revision history	Yes	Yes	Yes	Yes	Yes	Yes	No
Viewing revision properties	Yes	Yes	Yes	Yes	Yes	Yes	No
Editing revision comments	Yes	Yes	Yes	Yes	Yes	Yes	No
Merging revisions	No, except as part of merging views, which is often done by an administrator	Yes, using Visual Merge	No, except as part of merging views	No, except as part of merging views	No, except as part of merging views	No, except as part of merging views	No
Finding based on field content	No	Yes	Yes	Yes	Yes	Yes	Yes
Selecting by query See “Applying an Existing Query” on page 56.	No	Yes	Yes	Yes	Yes	Yes	Yes
Selecting by label See “Labeling an Item” on page 223.	No	Yes	Yes	Yes	Yes	Yes	No
Labeling revisions See “Labeling a Folder” on page 222 and “Labeling an Item” on page 223.	Yes	Yes	Yes	Yes	Yes	Yes	No
Viewing references See “Using References” on page 235.	Yes	Yes	Yes	Yes	Yes	Yes	No
Linking to folders and items See “Using Links” on page 227.	Yes	Yes	Yes	Yes	Yes	Yes	No
Printing This prints a default report for selected items	No	Yes	Yes	Yes	Yes	Yes	No
Sending items as e-mail	No	No	Yes	Yes	Yes	Yes	Yes

Table 14.1 Generic Operations for Folders and Items (continued)

Operation	Folder	File	Change Request	Requirement	Task	Topic	Audit
Receiving e-mail notification about changes (when notification is enabled by administrator)	No	No	Yes Changes in Responsibility only	Yes All changes in items for which you are responsible	Yes All changes in items for which you are responsible	Yes All changes in items for which you are a recipient	No
Controlling system tray notification See "Controlling System Tray Notification" on page 281.	No	No	Yes	Yes	Yes	Yes	No
Marking items as read/unread. See "Marking Items Read or Unread" on page 208.	No	No	Yes You can also mark trees as read/unread	Yes You can also mark trees as read/ unread	Yes You can also mark trees as read/ unread	Yes You can also mark trees as read/ unread	No
Flagging items	No	Yes	Yes	Yes	Yes	Yes	No
Deleting	Yes	Yes	Yes	Yes	Yes	Yes	No
Setting access rights. Normally, this function is performed by Administrators. See the *StarTeam Administrator's Guide.*	Yes	Yes	Yes	Yes	Yes	Yes	No
Creating reports. See "Reporting and Exporting Data" on page 249.	No	Yes	Yes	Yes	Yes	Yes	Yes
Creating charts. See "Using Charts" on page 259.	No	Yes	Yes	Yes	Yes	Yes	Yes

Moving Folders and Items Between Project Views

You can move folders and items from one project view to another as long as the two views use the same server configuration. You can also move them from one location to another in the same view.

When you move a folder, the folder's contents, including its child folders and the contents in the child folders, move along with the folder.

When an item (a file, change request, task, or topic) is moved, it belongs to the new project view. However, the items behavior, configuration and other properties do not change. The item does lose one property, its labels, meaning any labels it had in the previous view are lost. The reason for this is that labels cannot be moved from one view to another.

If a view is rolled back to an earlier point in time, it no longer shows the folders and/or items that have been moved.

Sometimes a disabled Branch On Change check box for a moved folder, file, or change request will become enabled because the moved folder or item has been propagated to other views. (Tasks and topics do not have branching behavior.)

Enabling the Branch On Change check box affects the workflow for change requests in the following ways:

- If the Last Build Tested and the Addressed In Build fields in a change request have build labels as their values (in other words, if these fields are not empty and do not contain the value “Next Build”), the moved change request retains those values.

  In the new view, these values can be changed, but only to the names of build labels that exist in the new view.

- If the Addressed In Build field contains the value “Next Build” at the time of the move, the “Next Build” value is replaced by the name of the next build label to be created in the original view—not the next build label created in the new view. This is true even if other changes have been made to this change request while in the new view.

- If the Last Build Tested and the Addressed In Build fields in a change request have no values at the time of the move, their workflow is specific to the new view only.

Note The comments about labels and workflow in the previous paragraphs in this section also apply to folders and items that appear in both a parent view and the branching view derived from it. However, the workflow of a change request is affected by its values in the Last Build Tested and the Addressed In Build fields at the time that the change request branches—rather than at the time it is moved.

An administrator is usually in charge of merging change requests and controlling which properties end up in which views during the merge process. However, merging change requests also produces the results explained above. For example, in the view in which the “Next Build” value was assigned, the next view label becomes the value of the Addressed In Build field, regardless of the view in which the merged change request now resides.

To move a folder or item between projects:

1. Select Project > Open from the menu bar to open first one and then the other view window. Make sure both project views are visible. You may need to resize the windows.

   If you are moving a folder or item from one location to another in the same view, you need only one view open.

2. Drag the folder or item to be moved from one project view to a folder in the other.

3. A message box asks you to confirm, click Yes.

Caution You cannot move tasks and subtasks that have been exported from MS Project to StarTeam.

Creating Shortcuts to Items

For easy access to items that you are tracking, you can save shortcuts to those items on your desktop. Opening the shortcut starts the application, opens the project view in the configuration it had when the shortcut was created, and displays the item’s *Properties* dialog.

To save a shortcut to an item:

1. Select a file, change request, topic, or task from the upper pane.

2. Select Save Shortcut from the item or context menu. The *Save As* dialog appears.

3. Enter a name or use the default name for the shortcut in the *File Name* text box. Do not change the .stx extension.

4. Select a location, usually your desktop, for storing this item shortcut.

5. Click Save.

To open an item with your shortcut, do one of the following:

- Double-click the shortcut from your desktop or Windows Explorer.
- Select Project > Open Shortcut from the menu bar. In the resulting *Open* dialog, select the shortcut name and click Open.

After you log on, the item's *Properties* dialog appears.

Opening and Copying StarTeam URLs

The Cross-Platform client can now open URL links to projects, views, folders, and items (files, change requests, requirements, tasks, and topics). By doing this, users can easily access specific locations in a project. Users can also copy a URL to the Clipboard, which allows them to easily move its contents to an appropriate application.

Like other URLs, StarTeam URLs include the name of the server for the connection. In some organizations, StarTeam servers may be reached from both the Internet and the corporate intranet. In such cases, a server may have two different IP addresses. If the user configures the server list to reference a server by its IP address, rather than its DNS name, then any URLs generated by the client will work only from the network on which that IP address exists.

URL displays may be changed in the user's Personal Options settings. For additional information see "Setting File Options" on page 275.

To open a URL:

1. From the Project menu, select Open StarTeam URL.
2. In the Open StarTeam URL dialog box, enter a valid URL to a project, view, folder, or item. For example: starteam://hostname:49201/myproject.
   - If the URL is a reference to a project, the default view of the project opens. If the URL is a reference to a view or folder, then the view or folder opens.
   - If the URL is a reference to an item, the item's view opens, the item's parent folder is selected in the folder tree, the item type is selected, and the item itself is selected in the item list or tree on the upper pane.

After opening a URL, users can copy it, then move its contents to an appropriate application. Copying a URL to the Clipboard is equivalent to dragging an item or items from the lists pane and folder tree onto an application.

To copy and paste a URL to the Clipboard:

1. Select one or more items.
2. From the item menu bar or context menu, select Copy URL to Clipboard. This action places on the Clipboard a plain text version of the URL to the selected items and an HTML representation of the links to the selected items.
3. From the Clipboard, paste the URL to a selected application.

Note Not all applications support pasting the HTML representation, although Word, Excel, and Outlook do support HTML data.

Sharing Folders and Items with Another View

Folders and items can be shared between views if the views belong to projects that use the same server configuration. When a folder is shared, users of both views can access its contents, including child folders and their contents. You can also share a folder or item at two locations in the same view.

Sharing folders is an important part of setting up a view. For example, suppose all products use a portion of a company's general libraries. Even though those libraries are not maintained by the developers of a given product, the product is based on some revision of source code and therefore must be compiled with these libraries. Therefore, some of the library folders should be included in the product's view.

When an item (folder, file, or change request) is shared in a new view, its behavior in this new view is controlled by a property called Set Items Shared Into View To Branch On Change. The item's behavior in the original view has no effect on its behavior in the new view. What's more, if the parent view is a reference view (one that does not permit branching), the new view does not inherit the properties of the parent view. In that case, the parent view's setting controls the shared item's Branch On Change setting.

Tasks and topics do not have branching behavior so this view property does not affect them.

The shared folder or item's configuration (floating, based on a label, a promotion state, or a point in time) is initially identical in both views. However, the configuration can be modified in either view. This means that the shared items can be different in each view.

The shared folder or item loses any labels it had in the previous view. Labels cannot be moved from view to view. However, the shared folder or item will have all the labels you attach to them in their new location regardless of whether they branch or not.

Caution When you share tasks that have been exported from MS Project, you must share an entire task tree, starting with the root task.

To freeze shared folders, configure each of them to a specific date and time. For more information, see "Reconfiguring Folders and Items" on page 199.

To share a folder or item between views:

1. From the menu bar, select Project > Open to open the project view, then open the other project view window. Make sure both project views are visible. If necessary, resize the windows.

   If a folder or item is shared from one location to another in the same view, only one view needs to be open.

2. Press *Ctrl* + drag the folder or item that is to be shared from one view to a folder in the other.

3. A message box asks you to confirm. Click Yes.

When sharing a folder or item, be aware of the following:

- Avoid circular shares. A circular share occurs when a folder or item, such as a file, is shared back into the original location. For example, you might share the same folder from the first view to a second view, from the second view to a third view and from the third view back into the first view. The share back to the first view makes this example a circular share.
- Deleting the original of a shared folder or item does not orphan the shared ones. Remember that nothing is ever really deleted from the application.
- Avoid redundant shares. A redundant share occurs when one share is identical to another share. For example, if you share the same folder from the first view into the second view more than once, you create redundant shares. What's more, the identical shared folders have files with the same names stored in the same working folders.

  Note Checking files out from one folder overwrites the files from the other and results in file status errors.

- Sharing a folder or item into a reference view also shares it into the parent view because the parent view was the source of the reference view. This is another example of a redundant share.

- If more than one third of a project consists of folders or items shared into the project, you should probably rethink the project's structure. It may be more complicated than necessary. Also, large amounts of sharing can cause some performance problems.

Understanding Branching

Understanding branching behavior is important because when a branching view (that is, a view that permits branching) is created, you must eventually decide whether to select the Branch All option or Branch None option. Your decision will affect every item in the new view that is derived from the parent view at the time the view is created.

Branching occurs when an item in the child view changes and has its behavior set to allow branching and to Branch On Change. When an item branches, a separation occurs between the item and its corresponding item in the parent view. These items also begin to have different branch revision numbers. When an item branches, the application no longer sends or applies updates from the corresponding item in the parent view.

When branching, note the following:

- If you select Branch All, the behavior of every item that is in the view at the time the view is created *is* set to Branch On Change.
- If you select Branch None, the behavior of every item in the view at the time the view is created is *not* set to Branch On Change. In other words, for any item with a floating configuration, changes to these items can be propagated to the parent view. (The changes might also be propagated to the corresponding items in some of the parent view's other child views—depending on what the corresponding item's current behavior and configuration is.)
- As you add, move, share, and modify items, their behaviors can change. The next few sections explain what particular behavior settings mean.

Note Any item with a frozen or fixed configuration is read-only when its behavior is *not* set to Branch On Change. Read-only means that no data about this item within the view can be changed. For example, although you may be able to edit a file, you cannot check it in or change its properties.

Branching Is Disabled

When the check box for Branch On Change is disabled, one of the following is true:

- The item has already branched.
- The item cannot be changed. For example, if the item is in a reference view, branching is not permitted. When such a view permits changes, the changes are automatically propagated to the parent view. No separation between the item and its corresponding item in the parent view ever occurs.
- Branching makes no sense for the item, given where and when it was created, moved, shared, etc. For example, a file that has just been added to a root view has no reason to be branched. An item that has no corresponding item in the parent view has no reason to branch.

  If you create a blank branching view (a view that is *not* derived from a parent view), no items are initially in the view. Therefore, no corresponding items exist in the parent view, so branching behavior is disabled.

Note In a branching view, when you add an item to a folder that does not branch (that is, the folder's Branch On Change check box is cleared), the new item has no corresponding item in the parent view. Because this new item is independent of the parent view, branching is disabled.

Branching Is Set to Branch On Change

When the Branch On Change check box is both enabled and selected, branching occurs the next time the item changes. At that time, a separation occurs between the item in the new view and its corresponding item in the parent view. The item that becomes separated from its corresponding item in the parent view takes on the following behaviors:

- Its Branch On Change check box becomes disabled
- Its revision number is new and not shared with a revision in the parent view

Branching Is Not Set to Branch On Change

When the Branch On Change check box is enabled but cleared, branching does not occur when you change the item.

If the item's configuration floats, the item's change is propagated to the parent view (and sometimes to other child views of the parent view).

If the item's configuration does not float, the item cannot be changed because the parent view cannot be updated. The item is treated as though it were read-only. Read-only means that the data of this item cannot be changed from within this view. For example, although you can edit a file, you cannot check it in or change its properties.

Branching a Folder, File, or Change Request

Suppose you are working on a product and a customer requests a special edition of the product. For example, the customer wants a couple of special features tailored specifically for them. To separate the items, such as files and change requests, for the current product from those for the special request, a branching view is created.

The folders selected from the parent view for the new view and the items found in those folders will be available in the new view. In fact, until an individual folder or item's behavior becomes Branch On Change and its properties are actually changed, the application stores only one folder or item for both views.

Note The branching behavior of a folder or item depends on its history. If you do not know the complete history of a folder or item, you should not assume that you know its behavior.

For example, if a folder or item was part of the parent view at the time the branching view in which it resides was created and if the branching view was created with Branch All as its branching option, the folder or item's branching behavior is initially enabled and the Branch On Change check box is selected. If the branching view was created with Branch None as its branching option, the folder or item's branching behavior is initially enabled and the Branch On Change check box is cleared. However, this behavior can be changed.

If a folder or item is added to the branching view after the view is created, the folder or item's branching behavior is disabled. The Branch On Change check box is cleared as well as inaccessible. On the other hand, if you share that folder or item, its branching behavior becomes enabled automatically in its new view.

A folder branches because its properties have changed. Branching a folder does not branch its contents (whether child folders or items). When a file branches, either its properties have changed or the file itself has been edited. When a change request branches, its properties have changed.

Items may also be branchable; that is, they can be derived from other items that become their ancestors. Items may have several completely different revision histories with common ancestries. In the case of a text file, for example, the branched item can

later be merged with the file from which it originated. For example, the development of a product for a new operating system may start with the existing files for the first operating system as its base.

When a folder, file, or change request branches, its revision number changes. For example, if a file's revision number is 1.13 before the file branches, the number becomes 1.13.1.0 after the item branches. The next change to that item in the branching view will have the revision number 1.13.1.1. The next change to that item in the parent view will have the revision number 1.14.

To review or change a folder or item's behavior:

1. Select the folder from the folder hierarchy or the item from the list in the upper pane.
2. Select Advanced > Behavior from the *Folder* menu, item menu, or one of their context menus.

   The *Item Behavior* dialog appears. The Branch On Change check box is enabled only if the folder or item can branch.
3. Do one of the following:
   - Select the Branch On Change check box so that the folder or item will branch with the next change.
   - Clear the box to keep it from branching with the next change.

   After branching takes place, this check box becomes disabled. The folder or item receives a new revision number that has two additional integers.
4. (Optional) If you are changing the behavior of a folder, you can change the behavior of its child folders simultaneously by selecting the Apply To Entire Folder Tree check box.

The folder's branching behavior affects items being added to the folder in the new branching view. Items can be added to the folder in a new branching view if:

- The folder has branched
- The folder has not yet branched but its behavior is "branch on change"
- The added item's behavior has the Branch On Change check box disabled.

If the folder has not yet branched and its behavior is *not* "branch on change", the file cannot be added; the error message says that the folder is read-only.

Changing a folder's branching behavior has *no* effect on the branching behavior of:

- Items in the folder before the change. Pre-existing items have their own behaviors.
- Items subsequently moved to or shared with that folder.
- Items subsequently propagated to that folder from parent views.

Reconfiguring Folders and Items

Individual folders or items can be reconfigured.

All configuration changes are set to either floating or to a particular date and time. For example, if you configure an item to a view label, the configuration of that item becomes either the date and time of the label or the date and time of the revision to which the label is attached (if the label has been moved). If you reopen the *Behavior* dialog and reselect the Configuration tab, the configuration is set to Revision As Of followed by the labeled revision's date and time.

If you reconfigure a revision to a promotion state, the configuration also becomes to the date and time, based on the label to which the promotion state was applied at the time of the reconfiguration. If you change the state so that it reflects another label, the folder or item's configuration does not reflect this change.

Folders and items can be:

- Rolled back from their existing date and time setting to an earlier view label, promotion state, or date and time.

  For example, you might want to check out a file as it was at a particular point in time and compare it with the tip revision.

  Rolling back a folder or item in the root view makes it read-only. Read-only means that no data about this folder item within the view can be changed. For example, although you can edit a file, you cannot check it in or change its properties.

  Rolling back a folder or item in a child view makes it read-only if it cannot branch.

- Rolled forward from one time in the past to a later, perhaps floating, time. The time might be represented by a a view label, promotion state, or the floating setting instead of a specific date and time.

  For example, when you create a branching view based on a specific label, promotion state, or time, each item in the child view has its configuration set to the corresponding time. If changes are made to the parent item before the child item branches, you can advance the configuration time to include the changes in the parent. For example, if an item is configured to June 5, 2003, 4:44 P.M. and changes are made to the parent item on June 8th and July 14th, you can advance the configuration time of the child item to an appropriate time on July 14th, thus including those changes in the child view. You can select several items at once and reset all their configurations to the same time simultaneously. This rebasing operation requires a confirmation.

A folder or item becomes read-only if it meets one of the following criteria:

- It is in a read-only reference view
- It is in a branching view and its behavior is either disabled or *not* set to Branch On Change

  Selecting Branch None as you create a new view makes every folder and item in the new view read-only until individual folders or items are set to Branch On Change or given floating configurations. Those folders or items will then either branch or the changes to them will carry over to the corresponding item in the parent view.

  For example, suppose some files are supplied to you by another team. You need to compile your files with theirs, but you do not want their files to be changed by the other team or by your team. You can roll back the configuration of these files to some point in time that they were stable and set their behaviors so that they cannot branch.

Unlike views, a folder or item keeps its configuration until you manually change the configuration or until the folder or item branches. After you close the view, the folder or item does not return to a status of current like a view does.

Changing a folder's configuration does *not* change the configuration of any of the items or child folders already associated with it. The reconfiguration only resets the folder's properties to the values they had at the configuration time. Depending on the folder's behavior, the folder may become read-only. When a folder is read-only, you cannot change its properties.

A folder's configuration does affect what new items or child folders can float into it. For example, to avoid having items float into a particular folder in a floating branching view, configure that folder to a label, promotion state, or point in the past. You can reconfigure that folder to floating later to receive new items from the parent. However, the items added to the parent while the folder was not floating will never automatically end up in the folder. They would have to be manually shared.

If you want to freeze a folder or item to a particular date and time, reconfigure the folder or item.

Other operations are better performed using less permanent solutions. For example, if you want to examine the contents of a specific file revision, such as the revision with the Beta label, select the file, then the Labels tab to find out what revision has that label. Lastly, select the History tab to check out that revision or compare it with the tip revision.

If you want to see what the contents of a view looked like at a particular point in time, reconfigure the view to that time. For example, to locate deleted folders or items, you need to roll back the view. Then you can see the folders, for example, that existed at the specified time even though they have since been deleted.

Caution There is no way to locate folders that have been configured to past times—unless you remember them. If they cannot branch, you can discover them because a read-only message appears when you try to change their properties. However, the folder might be read-only for another reason.

To reconfigure a folder or item:

1. Select the folder or item from the folder hierarchy.
2. Select Advanced > Behavior from the *Folder, item,* or context menu. The *<item> Behavior* dialog appears.
3. Select the Configuration tab.
4. Select a configuration option:
   - Floating

     This allows changes to the corresponding parent item to float into the child view (and sometimes vice versa). If this is a root view, it allows new revisions to be added to the item as the item is no longer read-only.
   - Labeled Revision

     This makes the folder or item revision with the view label that you specify the tip revision. This option is disabled if this view has no labels defined for it.

     The existing view labels are listed in reverse chronological order based on the time for which they were created.
   - Promotion State Configuration

     This makes the folder or item revision with the view label currently assigned to the selected promotion state the tip revision. This option is disabled if this view has no promotion states defined for it.
   - Revision As Of

     This makes the folder or item revision just prior to the specified point in time the tip revision.

     This option defaults to the current date and time, but you can select one in the past—so long as it is after the time at which the folder or item was created.

To identify rolled-back items:

1. Add the *Configuration Time* field to the upper pane (for details, see “Controlling the Columns” on page 53).

   Only items with configuration times in the past have any value is this field. You may want to sort based on this column. See “Sorting and Grouping the Data” on page 54.

   If the Configuration Time matches the time for a view label, the item may be configured to that view label or to a promotion state based on that view label.
2. To find the time associated with a view label, select View > Labels, select the label, then click Properties.

Other fields that may be of interest are Branch On Change and Branch State.

- Branch On Change indicates whether a file will branch when it changes. This is indicated by the values Yes or No. If Branch On Change is No and Configuration Time has a value, the item is frozen (read-only).
- Branch State indicates whether an item has branched in the child view, is still unbranched and, therefore, a part of the parent view, or is in the root view for the project (and, therefore, not branchable). Its values are Branched, Not Branched, and Root.

Locking and Unlocking Item

Whenever you plan to change a file's contents or edit an item's properties, you should exclusively lock that item. Locking an item alerts other team members of your intension to change that item and stops them from editing the item at the same time. An exclusive lock on an item prevents others from creating new revisions of that item before you release your lock. You can lock and unlock any type of item as separate operations, as explained in the following procedures. In addition, you can lock and unlock files as part of the check-in and check-out processes.

If the item is exclusively locked by another, you can review its properties but you cannot create a new revision of those properties. Normally the words "Read Only" and the name of the user who has locked the item appear on the title bar. However, if the item became locked after you opened it, you become aware of the lock only as you click OK.

To lock an item using the toolbar:

1. Select one or more items from the upper pane.
2. Click the Lock icon on the toolbar.

The selected items become exclusively locked by you.

To select a lock status for an item:

1. Select one or more items from the upper pane.
2. Select Lock/Unlock from the item or context menu. The *Set My Lock Status* dialog appears.
3. Select a lock status option:
   - Unlocked, to remove your exclusive or non-exclusive lock on the selected items
   - Exclusive, so that no one can create a new revision of this item except you (until you release the lock or someone breaks your lock)
   - Non-exclusive, to indicate that you are working on the item and may possibly make changes to the item (non-exclusive locks are not recommended for items other than files)
4. Depending on your access rights and privileges regarding a selected item, you may be able to break another team member's lock on that item. To break a lock, select the Break Existing Lock check box.

   If e-mail is enabled in the server configuration, the application sends e-mail messages to notify the team members whose locks you broke.
5. Click OK.

To remove your lock from an item, select Lock/Unlock or click the Unlock icon on the toolbar.

Depending on the settings for your personal options, items can be locked and unlocked automatically. See "Setting File Options" on page 275 and "Automatic Locking of Items Other than Files" on page 282.

Comparing Properties

You can compare the properties of two items of the same type. You can also compare the properties of two revisions of a folder or an item.

To compare the properties of two items of the same type:

1. Select the two items from the upper pane.
2. Do one of the following:
   - Click the Compare Properties icon on the toolbar.
   - Select Compare Properties from the item or context menu.

The *Compare* dialog appears and you can review the two sets of properties side-by-side. Differences are marked in bold typeface.

To compare the properties of two revisions of the same item:

1. Select the item from the upper pane.
2. Select the History tab.
3. Select the two revisions to be compared.
4. Do one of the following:
   - Click the Compare Properties icon on the toolbar.
   - Right-click the selected items in the *history* list and select Compare Properties.

   The *Compare* dialog appears and you can review the two revisions' properties side-by-side. Differences are marked in bold typeface.

To compare the properties of two revisions of the same folder:

1. Select the folder from the folder hierarchy.
2. Select Properties from the *Folder* or context menu.
3. Select the History tab. (For the History tab to appear, you must have access rights to use this tab and to view this folder's history.)
4. Select the two revisions to be compared.
5. Do one of the following:
   - Click the Compare Properties icon on the toolbar.
   - Right-click the selected items in the *history* list and select **C**ompare Properties.

   The *Compare* dialog appears and you can review the two revisions' properties side-by-side. Differences are marked in bold typeface.

Note Folder and item revisions can also be compared as part of a view comparison. Comparing and merging views is usually done only by administrators. It is explained in the *StarTeam Administrator's Guide*.

Using the History List

Selecting the History tab from the lower pane or from the *Folder Properties* dialog lists all the revisions of the selected folder or item. The history list contains several columns, each with its own column header. The columns for the history list are:

View	The name of the view to which the file belongs.
Revision	The number the application gave the revision as it was checked in.
Author	The person who checked in the revision.
Time	The date and time of the revision.

Comment	The reason provided for creating the revision.
Branch Revision	A special form of revision number that indicates the branch path for this revision.

The history list has its own context menu. Use it to:

- Compare a revision's properties with those of other revisions.
- Review a revision's properties or references.
- Add to or change the revision comment.
- For files, the menu also enables you to:
  - Check out a revision.
  - Review a revision's contents in the application associated with it.
  - Compare a revision's contents with that of other revisions or the working file.

Note Comparing the properties of two revisions is explained in "Comparing Properties" in the previous section. Reviewing revision properties and editing comments are explained below. The features that only apply to files are explained in "Checking Out a Revision" on page 136, "Reviewing the Content of a Revision" on page 140, and "Comparing File Contents" on page 140.

Reviewing a Revision's Properties

You can review the properties for revisions of an item or a folder.

To review properties for an item's revision:

1. Display the history list by doing the following:
   a. Select an item from the upper pane.
   b. Select the History tab below the view window's lower pane.
2. Select and right-click the revision in the history list. The History context menu opens.
3. Click Properties. The appropriate *Properties* dialog appears.

To review properties for a folder's revision:

1. Display the history list by doing the following:
   a. Select the folder from the folder hierarchy.
   b. Select Properties from the *Folder* or context menu.
   c. Select the History tab. (There is no History tab if you do not have the access rights that allow you to see this folder's history.)
2. Select and right-click the revision in the history list. The History context menu opens.
3. Click Properties. The appropriate *Properties* dialog appears. It does not display the Exclude and Files tabs.

Editing a Revision's Comment

As revisions of an item or folder are created (for example, by checking in a file or changing its properties), comments about each revision can be provided explaining reasons why changes were made. Later, additional comments can be added to or edited from the history list.

Caution Never use a keyword in a revision comment because it will be expanded during the keyword expansion process.

To edit the revision comment for an item's revision:

1. Display the history list by doing the following:
   a. Select an item from the upper pane.
   b. Select the History tab below the view window's lower pane.
2. Select and right-click the revision in the history list. The *History* context menu opens.
3. Click Edit Comment. The *Edit Comment* dialog displays the comment for the revision.
4. Edit the comment, then click OK.

To edit the revision comment for a folder's revision:

1. Select the folder from the folder hierarchy.
2. Select Properties from the *Folder* or context menu.
3. Select the History tab. (There is no History tab if you do not have the access rights that allow you to see a folder's history.)
4. Select and right-click the revision in the history list. The *History* context menu opens.
5. Click Edit Comment. The *Edit Comment* dialog displays the comment for the revision.
6. Edit the comment, then click OK.

Merging Revisions

Folder and item revisions can be merged as part of merging a view. However, merging views is usually done only by administrators. It is explained in the *StarTeam Administrator's Guide*.

Note You can also merge file revisions during the check-in or check-out process using Visual Merge. In addition, Visual Merge can be used as a stand-alone utility. For more information, see "Merging Files" on page 141 and "Using Visual Merge and Visual Diff" on page 111.

Finding Items by Field Content

You can search all the items displayed in the upper pane for the data contained in any displayed field. For example, you can locate a change request by its number or search for a file with a particular name, status, time stamp, or size. If you want to search a field that is not displayed, add that field by clicking the column headers then Show Fields.

To find items:

1. Select the File, Change Request, Topic, or Task tab from the upper pane.
2. Select Find from the item or context menu.

   The *Find* dialog appears.
3. Enter part or all of the data in the Search For text box. Do *not* use wild cards.

   If you are searching by number, be aware that the Cross-Platform client uses commas (or other numeric separators) while the Windows client does not. For example, to find change request 4344 on the Windows client, enter 4344. To find the same change request on the Cross-Platform client, you must enter 4,344.
4. Select Forward to search the upper pane from the top to the bottom or select Backward to search the upper pane from the bottom to the top.

5 Select Starting At Currently Selected Item to begin searching from the item that is presently selected or select First Item or Last Item to search from beginning or end of the list (depending on whether you selected a forward or backward search).

6 Select either the All Displayed Fields or This Field option. If you select This Field, select the field you want to search from the drop-down list box.

7 Select the Match Case text box, depending on whether a case-sensitive search is appropriate.

8 Click OK to search.

Tip For greater efficiency, use the shortcut keys:

- *Ctrl+F* to start a search
- *F3* to find the next change request that matches the search text
- *Shift+F3* to find the previous change request that matches the search text

Printing Items

You can print the data that appears in the upper pane as a default report. However, the Print command works only if your computer has a “print” action file association for the .htm extension. Installing a browser on your computer creates this association.

To print the data displayed in the upper pane:

1 Make sure the rows and columns of information you want appear in the upper pane.

2 Select the rows to be printed.

3 Select Print from the File, Change Request, Topic, Task, or Audit menu or one of their context menus.

The application prepares a default report on the selected rows of items and prints that report.

Sending Items via E-mail

You can send a text representation of the selected item properties as an e-mail message along with some additional text. The information sent for each item includes the fields displayed in the upper pane. For example, if you select a file, you can send its properties, but not its contents. If you select a change request, the properties are its “contents”.

Items sent via the application are from the application, not from you. If you want to send a copy to yourself, select the Send Copy to Myself check box. Otherwise only the addressees receive the e-mail.

If only one recipient of a sent item or notification has an e-mail address that is incorrectly formatted (as in “junk”), no one receives the e-mail.

If an e-mail address is invalid, but formatted correctly (as in “junk@place.com”), the e-mail is sent to all valid recipients, and the sender gets an “Undeliverable message” from the e-mail system for the invalid address.

To send item properties via e-mail:

1 Select one or more items from the upper pane.

2 Do one of the following:

- Click the Send icon on the toolbar.

  (Be aware that the File component does not have a Send button on its toolbar.)

- Select Send To from the item or context menu.

The *Send To* dialog appears. This dialog has more options in the Cross-Platform client than in the Windows client.

3 To address the e-mail to team members:

a Click To.

The *Select Primary Mail Recipients* dialog appears.

b Select recipients from the *Available Users* list box, then click Add to place them in the *Selected Users* list box. The names in the *Selected Users* list box will appear in the To text box.

c Click OK to return to the *Send To* dialog.

Notes

You can enter addresses of users directly into the To or Cc text boxes as long as they can be evaluated uniquely by your mail application. For example, if two people have jmarsh as the beginning of their e-mail addresses or if jmarsh has two e-mail addresses known to the mail application, this cannot be resolved uniquely. So jmarsh is not a sufficient address.

When you select recipients from the Select Primary/Secondary Mail Recipients dialogs, the mail might still be undeliverable because the e-mail address listed for a team member (as one of his user properties) is ambiguous or incorrect.

4 To carbon-copy other team members:

a Click Cc. The *Select Secondary Mail Recipients* dialog appears.

b Select recipients from the *Available Users* list box, then click Add to place them in the *Selected Users* list box. The names in the *Selected Users* list box will appear in the Cc text box.

c Click OK to return to the *Send To* dialog.

5 Enter a subject in the *Subject* text box.

6 (Optional) To receive a copy of this e-mail, select the Send A Copy To Myself check box.

7 (Optional) To send a shortcut to an item via e-mail, select the Attach Item Shortcut check box.

Note

An item shortcut starts the application, opens the project view in the configuration it had at the time the shortcut was created, and opens the item's *Properties* dialog. If more than one item is sent at the same time, only one shortcut appears in the e-mail message. When the shortcut is opened, you can move from one sent item to another using the *Properties* dialog's Next and Previous buttons.

8 (Optional) To send any files attached to this item, select the Include Item Attachments check box.

9 (Optional) To send the properties of the selected items as an HTML page instead of as the contents of an e-mail message, select the Send Items As HTML Attachment check box. The body of the e-mail message will be empty.

10 (Optional) Enter additional information in the A*dd Text To The Mail Message* text box.

11 Click Send Now to send the message.

Note

Do not confuse e-mail messages sent by individuals with e-mail notification messages automatically sent by the Server. If your administrator has enabled e-mail notification, you will automatically receive e-mail messages about change requests for which you are responsible or any changes in requirements, tasks, or topics for which you are responsible or the recipient.

Controlling System Tray Notification

Some personal options control how often you are alerted about new items using system tray notification. See “Controlling System Tray Notification” on page 281. In the Windows client, notification icons appear in the system tray portion of the task bar. In the Cross-Platform client, notification icons appear on the status bar.

Marking Items Read or Unread

It is sometimes convenient to mark items as read or unread. Because tasks and topics can be displayed in a tree format as well as a list format, you can also mark an entire tree (called a thread) as read or unread.

To mark items as read or unread:

1 Select the items to be made bold (unread) or not bold (read).

2 Select Mark as Unread or Mark as Read from the item or context menu.

To mark an entire thread as read or unread:

1 Click the Tree Display button to display the tasks or topics as trees.

2 Select the root of a tree.

3 Select Mark Thread as Unread or Mark Thread as Read from the item or context menu.

The entire tree or thread is marked read or unread.

Managing Read/Unread Items

Some personal options control when items become marked as read or unread. See “Managing Read/Unread Items” on page 280.

Flagging Items

You can flag items (except for audit entries) as a way of identifying items in the item list. Flags are set, viewed and removed by the user who created them. For example, you may want to use flags to remind you to follow up on a customer request. The Flag field displays either Yes and a blue flag indicating that the item is flagged or No indicating the item is not flagged.

To flag an item:

1 Select the item you want to flag.

2 Select Flag from the item or context menu.

The item list now displays the flagged items.

To view flagged items:

1 Right-click a column header.

2 Select Show Fields from the context menu.

3 Select the Flag field from the Available Fields list.

4 Click Add.

To clear a flag:

1 Select the item you want to clear a flag from.

2 Select Remove Flag from the item or context menu.

Tip For greater efficiency, use the shortcut keys:

- *Ctrl+Shift* to flag an item
- *Ctrl+Shift+F2* to remove a flag from an item

Deleting a Folder or Item

Your privileges and access rights determine whether you can delete a folder or item.

When you delete a folder or item, it disappears from the view from this time forward. Rolling back the view returns the folder or item to visibility. Use this procedure with caution.

Deleting a folder from a project deletes all the items associated with that folder. It also deletes the folder's child folders and their contents. However, it does not delete the folder's working folder or any files in that folder.

You cannot delete the root folder from a view. To get rid of a root folder, you must delete the view.

Deleting a topic or response also deletes its child responses, that is, the entire branch for which the selected topic or response is the root.

Deleting a task or subtask also deletes its subtasks, that is, the entire branch for which the selected task or subtask is the root.

If you delete a file, you remove it from version control and can, optionally, also delete the working copy of this file from its working folder.

To delete a folder:

1 Select the folder from the folder hierarchy.

2 Select Delete from the Folder or context menu.

The message, "Delete Folder *Name*?" displays.

3 Click OK.

To remove one or more files from version control:

1 Select the appropriate folder from the folder hierarchy.

2 Select the File tab.

3 Select the files to be removed from the File list in the upper pane.

4 Select Delete from the File or context menu. A message box asks if you want to delete the corresponding working files from the working folders on your computer.

5 Decide whether the working files should be deleted, then select Yes, No, or Cancel.

6 Confirm the deletion of each file from the application. After deleting a file, you will no longer see information on the file in the current view. However, this information remains in the project repository and the archive (or vault) until the file is purged. To access a deleted file that has not been purged, you can roll back a folder or view to a time when the file existed.

If a file has the status Not In View, deleting it removes the copy on your workstation. It does not delete the file from the application because it is not currently in the application.

To delete items (except files and audit entries):

1 Select an item from the upper pane. (You can select more than one change request.)

2 Select Delete from the item or context menu.

One of the following messages displays:

- “Delete Change Request #?”
- “Delete Requirement # and its sub-requirements?”
- “Delete Task # and its sub-tasks?”
- “Delete Topic # and its sub-topics?”

3 Click OK to confirm the deletion.

Chapter

15

Using Process Rules

Modern development practices require increased control over the entire development process. The application's process rules take change management to a higher level by requiring developers to follow a defined development process, one that ensures that all changes be linked to either a change request, requirement, or task. Items used in this way are known as process items.

Unlike typical file-based software configuration management (SCM) systems, the process rules integrate change management with SCM. With process rules, users can clearly distinguish which file revisions resolve a change request, complete a requirement, or finish a task. This ensures that code and content are modified only as the result of a clearly defined and approved objective, one expressed in the change request, requirement, or task.

Process rules are useful when creating baseline builds or configurations. A build is a labeled configuration that identifies the file revisions and other items that define the code and content baseline. Without process rule enforcement, developers using the application create baselines either by:

- Labeling an entire project view at a point in time
- Associating file revisions with a revision label on check-in

Even without process rules enforced, a user can select a change request, requirement, or task to use as a process item when adding or checking in files. In addition, the user can mark the change request as fixed the requirement as complete, or the task as finished as part of this process. If you have selected a disabled requirement (which is read-only because it was imported from CaliberRM) as a process item, it cannot be marked complete.

As a convenience, the user can select a change request or a task as the Active Process Item prior to adding or checking in files.

Using Process Items

Whether or not process rules are required, you can take advantage of the linking and tracking made possible with process items. As you add files or check them in, you indicate that the new file revisions are to be linked and pinned to a specific process item. You do this by selecting a change request, requirement, or a task as the process

item for the operation. At the same time you can mark the change request as fixed, the requirement as complete, and the task as finished.

Using process items enables you to clearly distinguish the following:

- Which file revisions are related to or fix a specific change request
- Which file revisions are related to or complete a specific requirement
- Which file revisions are related to or finish a specific task

To specify a process item, you can:

- Pre-select it from the main window.

  For example, after reading a change request for which you are responsible, you might decide to change its status to In Progress and work on it immediately. You can use a menu command or toolbar button to designate that change request as the active process item. See "Pre-selecting an Active Process Item" on page 212.
- Select a process item from the *Add Files* or *Check In Files* dialog. See "Adding Files to a View" on page 130 and "Checking In Files" on page 134.

Understanding Process Item Restrictions

Only change requests, requirements, and tasks can be process items. If process rules are *not* enforced, you can use any change request, requirement, or task as a process item, regardless of its status.

If process rules are enforced, you may be limited to selecting only one type of item as a process item. For example, you might be required to select only change requests. In addition, you might be limited to selecting items with specific statuses. If you do not have the access rights to review properties, ask your administrator what restrictions have been placed on process items.

Process item selections are per view. As you change from view to view, you will see the process item information displayed on the status bar change.

To review project properties:

1. Select Project > Properties from the menu bar.

   The *Project Properties* dialog appears.
2. Select the Process Rules tab.
3. Review the restrictions. If no statuses are selected for permitted items, then any status can be used.

If you plan to use process rules, you must have the correct access rights, which are:

- The rights to see and modify items used as process items in the project view
- The rights to create and modify links on files and items used as process items

Pre-selecting an Active Process Item

Selecting an active process item is a convenience that can save you time as you add files or check them in later. For example, after you review a task, you usually know if you are going to work on it next or not. Because you already have the task selected from the upper pane, making that task the active process item is a simple operation.

Only one item can be specified as an active process item at a time.

To pre-select an active process item:

1 From the upper pane, select an appropriate item.

When process rules are enforced, you must select a permitted type of item with a permitted status.

2 Do one of the following:

- Select Change Request > Set Active Process Item.
- Select Requirement > Set Active Process Item.
- Select Task > Set Active Process Item.
- Click the Set Active Process Item icon on the toolbar.

Although a Clear Active Process Item command appears on the Change Request, Requirement, and Task menus, you will probably never use it. Selecting a second active process item clears the first.

You do not have to use the active process item while adding files or checking them in. The active process item becomes the default selection for a process item, but you can select another appropriate item.

Linking Files to Process Items

When you add files to a view or check them in, you select the process item to which your files will be linked. One end of the link will be pinned to a new file revision and the other end will be pinned to the current revision of the change request, requirement, or task.

As part of the add files or check-in process, you can change the change request's status to Fixed, the requirement's status to Complete, or the task's status to Finished. If the process item becomes fixed, complete, or finished, each new file revision is pinned to the new revision of the process item, the one with the correct status.

Using the Links to a Process Item

You can determine what file revisions are associated with a process item by reviewing the links for that item.

To find the file revisions linked to a process item:

1 Select the process item from an application client's upper pane.

2 Select the Link tab from the lower pane.

3 Do one of the following:

- Click the Item Type column header to sort the links and more easily review the links from the selected process item to files.
- Right-click the column headers and use the *Sort and Group* dialog to find links to files.

You can manually link one item to another more than once. However, if the application links a file to the same process item through multiple add and check-in operations, the application updates the existing link rather than creating a new link. That means that, no matter how many revisions of the same file were pinned to revisions of the same process item, the last revision of that file to be pinned to the process item is the only one specified in the link's properties.

4 Right-click a link in the Link pane and select Link Properties from the context menu.

The *Link Properties* dialog appears.

5. Examine the *Link Properties* dialog to find the information that you need.

   Each link has two ends. The item selected from the upper pane is at one end and is called the source item. The other item is called the target item.

6. Repeat steps 4 through 5 for the other links.

Chapter

16

Using Labels

The application allows you to create two types of labels:

- View labels, which generally speaking “time stamp” the entire contents of the view. When you roll back the view to that label, you see everything that existed at that point in time—unless the label has been adjusted.

  A view label is created for a specific point in time or as a copy of another existing view label. Unless the view label is frozen, you can adjust it to include or exclude some folders and items. You do this by attaching or detaching the view label from them. You can also move a view label from one revision to another. For example, a couple of files might not have been checked in prior to the creation of the label but need to be included.

- Revision labels, which are not attached automatically to any item in the view.

  The purpose of a revision label is for use with a set of folders or items within the view. For example, you might want to label a group of files that should be checked in and out together.

Every label name is unique within its view. In a view, no view label can have the same name as any other view label; no revision label can have the same name as any other file revision label; and no view label and revision label can have the same name.

Any item, for example, a file revision or change request revision, can have more than one label. However, no two revisions of the same item can have the same label at the same time. Any label can be detached from a folder or item, leaving no revision with that label.

You can create and manipulate labels only while the view has a current configuration. No labels can be created, moved, and so on in a rolled back view.

Either type of label can be attached to or detached from a folder or item manually using the Labels command on the Folder menu, item menu, or one of their context menus. In addition, you can use a label (of either type) to identify files as they are checked out. You can attach (or attach and create) a revision label as you check a file in.

You can move a label from one revision to another. For example, suppose old help files were used in the build because the new help files were not ready. As soon as the new revisions of the help files are checked in, you can move the view label to the new revisions.

For another example, suppose you check in several files simultaneously, giving them a label. Then you decide that you need to make one more change to one of them, resulting in a new revision of that one file. When you check in the new revision, you can add it to the group by giving it the same label as the group. However, that detaches this particular file revision label from the previous revision as no file ever has more than one revision with the same revision label.

You can select any type of item (file, change request, task, or topic) by label. For example, you can select all the files that have a particular revision label. After they are selected, you can roll these files back to that label (which makes the revision with that label the tip revision). Then you can compare the working files to their rolled back revisions.

Creating View Labels

View labels have a variety of uses. Using the view label as a time stamp, you can roll back a view to any view label to see everything as it was at a specific point in time. For example, to see if a particular file was in the beta version of a product, you can roll back the view to the beta label.

By using view labels you can let testers know what build to use when verifying a change request if the following are true:

- The view label has been designated as a build label.
- The build label is created while the change request's Addressed In field contains the value "Next Build". In this case, the build label's name replaces "Next Build" so testers know which build to use.

For example, suppose your administrator creates a view label before each build, giving that label to the tip revision of each file in the view, and then checks out all the files with that label for the build. If the tip revision of one file does not change for a few weeks (or longer), it can acquire several view labels, while a file that changes frequently may have several revisions with no view labels and other revisions with only one view label.

When you detach a view label from a folder, the label is automatically detached from everything in the subtree for which the folder is the root. If you roll back a view to a specific view label and a folder does not have that label, you cannot see that folder's children and their contents anyway.

A view label can be created only at the view level and only while the configuration is current. However, it can be created for the current configuration or for a time in the past. In either of these two cases, the new label is attached to the tip revision of each folder, file, change request, task, or topic that belonged to the view at the specified time.

A view label can also be created as a copy of an existing view label or as a copy of the view label currently attached to a promotional state. In these two cases, the new label is attached to exactly the same items and revisions as the existing view label.

You can check file revisions out using this label or roll back the view to this label and see all the items associated with that label. For example, if you create the view label Build 100 as you make Build 100 of your product from a view, all the files in the view will have the label Build 100.

If some items should not be included, you can detach the label from those items individually. For example, if some files should not have that label, select the files then select Labels > Detach from the File menu or context menu to detach that label. If the files that should not be included all belong to the same folder and are the only files in that folder, use the Labels command on the Folder menu. For example, if the help files were not checked in until after the view label was attached, you can move that label from the previous revisions of the help files to the newly checked-in help files.

A view label can be designated as a build label. When you use build labels, the QA team can immediately determine what build to test for a fix to any given change request. This is because, when the application creates the build label, each change request that has “Next Build” as the setting for its Addressed In Build property is reset to the build label instead of “Next Build”. (Any change request that has “Next Build” as the setting for its Addressed In Build property has been resolved since the previous build label was created.)

To create a new view label:

1. Select View > Labels from the menu bar.

   The *Labels* dialog appears with the View tab selected. The existing labels are listed (along with their descriptions) in reverse chronological order based on the time for which they were created.

2. Click New to create a new label and add its name to the list box. The *View Label* dialog appears.

3. Enter a name and description for the label in the appropriate text boxes. The maximum label name length is 64 characters and the description length is 254 characters.

4. Select one of the following option buttons:

   - Current Configuration, to attach the label to the tip revision of every item in this view’s current configuration.
   - Labeled Configuration, to attach the label to the revision with the label you specify. The existing view labels are listed in reverse chronological order based on the time for which they were created.

     Creating a view label based on another view label is equivalent to copying that view label.
   - Promotion State Configuration, to attach the label to the revision currently in the promotion state that you specify (actually, the label is attached to the revision that also has the promotion state’s current view label).
   - Configuration As Of, to attach the label to the revision that was the tip revision at the specified date and time.

5. (Optional) Select the Use As Build Label check box to update each change request that has “Next Build” as the setting for its Addressed In Build property. If this option is not selected, change requests will still have this label, but it will not affect the setting of the Addressed In Build property.

6. (Optional) To freeze this label so that the revisions attached to it cannot be changed, select the Frozen check box.

Note The computers that run the Server and client must have their dates and times synchronized. Many features of the application depend on calculations involving times and dates; in particular, labels, configurations, and promotion states are all governed by time and date calculations. If the client and server are not kept synchronized, a number of operations may fail, or produce inaccurate or unreliable operations such as checkout, file status displays, or label creation.

Copying View Labels

Occasionally, users want to create a view label that is attached to exactly the same revisions of items as an existing view label, with a few exceptions. For example, suppose the files with the view label Release 1.0 are shipped as a product. Later, a patch is created to fix a problem with the new product. The source code files for the patch have the revision label Migration Patch. You want to create a new view label named Release 1.0 and Migration Patch. However, when you create a view label, it

becomes attached to the tip revisions of all the current files. Not all of these files were used to create Release 1.0, and, of the files that were used to create Release 1.0, the revisions used in the released product are no longer tip revisions.

You can copy view labels if they exist in the view where you are performing the operation and if the view's configuration is current. You *cannot* copy a view label from another view.

To create a view label based on an existing view label with a few exceptions:

1. Create a new view label attached to the same revisions of items as the existing view label.
   a. Select View > Labels from the menu bar. The *Labels* dialog appears with the View tab selected.
   b. Click New. The *View Label* dialog appears.
   c. Enter a name, such as Release 1.0-Patch 1, and a description for the label in the appropriate text boxes.
   d. Select the Labeled Configuration option button to attach the label to the revisions with an existing label, for example, Release 1.0.
   e. (Optional) By default, all view labels are designated as build labels. If you do not want this label to be a build label, clear the Use As Build Label check box.
   f. (Optional) You can freeze this label by selecting the Frozen check box.
   g. Click OK. Your new view label is now attached to the same revisions as the existing label.
2. Select the files for which the new label must differ from the copied label. For example, select all the files used to make the patch (and, hopefully, checked in with a specific revision label).

   If these files do not have a revision label, select them individually instead of following the steps below.
   a. Choose Select > By Label from the item or context menu. The *Select A Label* dialog lists all the existing labels.
   b. Select the revision label associated with the selected files, for example, Migration Patch.
   c. Click OK. Files that have this label become selected in the upper pane.
3. Move your new label to the appropriate revisions.
   a. Select Labels > Attach from the File or context menu. The Attach a Label dialog lists all the existing labels.
   b. Select your new label from the list (for example, Release 1.0-Patch 1).
   c. Select the Labeled Configuration option button to adjust your new label to include the revisions in the patch.
   d. Select the revision label that is associated with the patch (for example, Migration Patch) from the drop-down list box.
   e. Click OK.

   For the files selected from the upper pane, your new view label is moved to (or added to) the revisions that have the selected revision label. Your new view label now differs slightly from the view label you copied.

   If these files do not have a revision label but are the tip revisions, select the Current Configuration option button instead of the Labeled Configuration option button. If they are not tip revisions, perhaps you can use the Configuration As Of option button.

4 Repeat steps 2 and 3 if you need to adjust the new view label for types of items other than files.

Creating Revision Labels

A revision label can be created at the view level, as part of the file check-in process, or as part of the attachment process for folders or items. However, if you are copying a new revision label from an existing revision label in another view, you must use the View menu.

As you move or share an item from one view (the source view) to another (the target view), the labels from the source view do not become part of target view. However, you can move or share the items into the target view and then copy the revision labels. This enables you to selectively maintain revision labels on the moved or shared items.

For more information about creating and attaching labels as you check them in, see "Checking In Files" on page 134. They can also be created as you attach them to a folder or item revision. See "Labeling a Folder" on page 222 and "Labeling an Item" on page 223.

You can also check files out based on a label. For details, see "Checking Out a Revision" on page 136.

To create a new revision label:

1 Select View > Labels from the menu bar. The *Labels* dialog appears.

2 Select the Revision tab.

The *Labels* dialog lists the existing revision labels (along with their descriptions) in reverse chronological order based on the time at which they were created.

3 Click New to create a new label and add its name to the list box. The *Revision Label* dialog appears.

4 Enter a name and description for the label in the appropriate text boxes. The maximum label name length is 64 characters and the description length is 254 characters.

5 (Optional) To freeze this label so that the revisions attached to it cannot be changed, select the Frozen check box.

6 Click OK to exit the *Revision Label* dialog.

Copying Revision Labels

Occasionally, you will want to copy a revision label. For example, suppose you create a patch for a released product using the revision label Migration Patch to label the files as they are checked in. Suppose the patch is sent to customers, but, unfortunately, another change needs to be made to one of the source code files used to create the patch. To differentiate between the two patches, you can create a new revision label, for example, Migration Patch 2, based on the revision label Migration Patch. Then you can adjust Migration Patch 2 to include the changed source code file.

You can copy a revision label in a variety of ways. This section shows the method that enables you to copy a revision label whether it exists in the current view or in another accessible view.

As you copy a revision label, it immediately becomes attached to the same revisions of the same items as the original revision label. If the two revision labels exist in the same view, each will be attached to the same number of items.

When you copy a revision label from another view, the new revision label becomes attached to the same revisions of the same items, but only if these items (and their

correct revisions) exist in the new revision label's view at the time of the copying. For example, if the two views share only a few items or if items have been moved from one view to the other, the new revision label may be attached to a different number of items than the original revision label is attached to.

The following procedure assumes that you are dealing with files, but the same procedure can be adapted for other items.

To create a revision label based on an existing label with a few exceptions:

1. Create a new revision label attached to the same revisions as the existing label.
   a. Select View > Labels from the menu bar. The *Labels* dialog appears.
   b. Select the Revision tab.
   c. Click New. The *Revision Label* dialog appears.
   d. Enter a name, such as Migration Patch 2, and a description for the label in the appropriate text boxes.
   e. Select the Copy From Another Revision Label check box.
   f. Click Select. The *Copy a Revision Label* dialog appears.
   g. Select the project view that contains the revision label to be copied using the Project list box and the View tree.
   h. Select the revision label from the Labels list box.
   i. Click OK to exit the *Copy a Revision Label* dialog.
   j. Click OK. Your new revision label is now attached to the same revisions as the existing label.
2. Do one of the following:
   - Check in the changed file or files using the new revision label.
   - Attach the new revision label manually to the changed file revisions.
     1. Select the changed file.
     2. Select the Label tab beneath the lower pane.
     3. Drag the new revision label to the correct (probably tip) revision.
     4. Repeat the steps above for any other changed files.

If you have added a file rather than changed a currently labeled file, you must use the Labels > Attach command instead of dragging-and-dropping the label from one revision to another.

Freezing Labels

If you freeze a label, no one can add it to or remove it from any revisions.

To freeze or unfreeze a label:

1. Select View > Labels from the menu bar. The *Labels* dialog appears.
2. Select the View tab if the label to be frozen or unfrozen is a view label. Otherwise, select the Revision tab.
3. Select the label from the list box.

   The existing labels are listed (along with their descriptions) in reverse chronological order based on the time for which (in the case of view labels) or at which (in the case of revision labels) they were created.
4. Click Freeze or Unfreeze.

The Freeze/Unfreeze button changes its name depending on the current status of the selected label. For example, if the label is currently frozen, the button reads Unfreeze.

You can identify a frozen label by an icon with a small snowflake displaying in its upper right corner. The icon displays in front of the label name in the list box.

When a label is frozen, the label cannot be:

- Attached to additional folders or items
- Detached from any folder or item
- Moved from one revision of a folder or item to another

Deleting Labels

Deleting labels detaches them from revisions and removes them from the database. If the label is frozen, you must unfreeze it before you can delete it.

To delete a label from the current view:

1. Select View > Labels from the menu bar. The *Labels* dialog appears.
2. Select the View tab if the label to be deleted is a view label. Otherwise, select the Revision tab.

   The existing labels are listed (along with their descriptions) in reverse chronological order based on the time for which (in the case of view labels) or at which (in the case of revision labels) they were created.
3. Select the label from the list box.
4. Click Delete.

The label is deleted from the view. It is no longer visible in any list of labels. It is no longer attached to any folders or items—unless you roll back the view to a date and time when the label existed. However, you can create a new label with this exact name later.

Reviewing Label Properties

A label's properties include its name, description, and frozen/unfrozen status. A view label's properties also include its configuration date and time and whether it is a build label. You can also freeze/unfreeze a label from its properties dialog.

To review a label's properties:

1. Select View > Labels from the menu bar. The *Labels* dialog appears.
2. Select the View tab or the Revision tab.

   The existing labels are listed (along with their descriptions) in reverse chronological order based on the time for which (in the case of view labels) or at which (in the case of revision labels) they were created.
3. Do one of the following:
   - Double-click the label.
   - Select the label from the list box, then click Properties.

A properties dialog appears, listing the label's properties.

Labeling a Folder

Labeling folders is slightly different than labeling items. For example, you use a menu command rather than the Label tab to see the labels attached to a folder.

When you attach a label to a folder, you can also attach it to any of the following:

- The items that the folder contains
- The folder's contents, child folders, and their contents (everything in the subtree for which the folder is the root)

When you attach a label to an item or a folder, that label is automatically attached to all of the item or folder's parent folders in the view. If the label is a revision label, you can also detach the selected revision label from this folder and the items associated with it and, optionally, detach the label from or keep it with the folder's child folders and their items.

When you detach a view label from a folder, the label is automatically detached from the items that the folder contains. It is also automatically detached from the folder's child folders and their contents. If you roll back a view to a specific view label and a folder does not have that label, you cannot see that folder's children and their contents anyway.

To review the labels attached to folder revisions:

1. Select the folder from the folder hierarchy.
2. Select Labels from the Folder or context menu.

   The *Labels* dialog lists the labels currently attached to this folder or item on a revision-by-revision basis.

To move a label from one revision to another:

1. Select the folder from the folder hierarchy or the item from the upper pane.
2. Select Labels from the Folder or context menu. The *Labels* dialog appears.
3. Drag the label from one revision to another.

To create a revision label and attach it to a folder and its contents:

1. Select the folder from the folder hierarchy or the item from the upper pane.
2. Select Labels from the Folder or context menu. The *Labels* dialog appears.
3. Select a revision to attach the new label to.
4. Click New to create a new revision label. The *Attach a New Revision Label* dialog appears.
5. Enter a name and description for the label in the appropriate text boxes. The maximum label name length is 64 characters and the description length is 254 characters.
6. (Optional) To ensure that only the selected revision can have this revision label, select the Frozen check box.
7. Do one of the following. To attach the label to:
   - Only the selected folder, select the Folder Only option button.
   - The folder and its non-folder contents, select the Folder And Items Contained In Folder option button.
   - The folder, its contents, its child folders and their contents, and so on, select the Everything In Subtree Rooted At Folder option button.

Note Because attaching a label to a folder can also attach the label to the folder's contents, children, and so on, the label is always attached to the current

configuration of each folder and item. When you label a specific item, you can label a past revision of it, but with folders this is not possible.

To attach an existing label to a folder and its contents:

1. Select the folder from the folder hierarchy.
2. Select Labels from the Folder menu, item menu, or one of their context menus. The *Labels* dialog appears.
3. Click Attach.

   The *Attach a Label* dialog lists all the existing labels and identifies them as view or revision labels. By default, both types of labels are listed because both the View Labels and Revision Labels check boxes are selected.
4. (Optional) You can limit the list to only view labels or only revision labels by clearing the check box for the type of labels you do not want to display.
5. Select a label from the list.
6. Do one of the following. To attach the label to:
   - Only the selected folder, select the Folder Only option button.
   - The folder and its non-folder contents, select the Folder And Items Contained In Folder option button.
   - The folder, its contents, its child folders and their contents, and so on, select the Everything In Subtree Rooted At Folder option button.

To detach a label from a folder and its contents:

1. Select the folder from the folder hierarchy.
2. Select Labels from the Folder or context menu.

   The *Labels* dialog lists the labels currently attached to this folder.
3. Select the label to be detached from this folder.
4. Click Detach. The *Detach Label* dialog appears.
5. (Optional) If you are detaching a revision label from a folder, select the Folder Only, Folder And Items Contained In Folder, or the Everything In Subtree Rooted At Folder option button.

To display a label's properties, either double-click the label or select the label, then click Properties. A revision label has a name and a description. A view label has a name, description, and a configuration time.

Labeling an Item

In addition to the operations available when you label folders, you can select items from the upper pane based on whether they have a specific label. Then, for example, you can print reports about them. If you flag the selected items, you can create a query to find them later.

The Label pane displays a tree of the selected item's revisions showing the labels attached to those revisions as the children of those revisions. A label's type (view or revision) precedes its name in the tree.

Right-clicking a revision displays a context menu from which you can attach another label to the revision. When you right-click a label, you can review its properties and detach it from its revision. Whatever you right-click, you can fully expand or completely collapse the tree of revisions and labels by selecting Expand All or Collapse All from the context menu.

Double-clicking a revision expands or collapses it as well. Double-clicking a label displays its properties.

To review the labels attached to item revisions:

1 Select the item from the upper pane.

2 Select the Label tab.

The labels for the selected item display in the lower pane in much the same way they display in the *Labels* dialog (for folders) above.

To move a label from one revision to another:

1 Select the item from the upper pane.

2 Select the Label tab.

The revision names and the label for each revision display in the lower pane.

3 Drag the label from one revision to another.

To create a revision label and attach it to an item:

1 Select the item from the upper pane.

2 Select Labels > New from the item or context menu. The *Attach a New Revision Label* dialog appears.

3 Enter a name and description for the label in the appropriate text boxes. The maximum label name length is 64 characters and the description length is 254 characters.

4 (Optional) To ensure that only the selected revision can have this revision label, select the Frozen check box.

5 Indicate what revision of the item is to receive this label by selecting a configuration option button. Your choices are:

- Current Configuration, to attach the label to the tip revision.
- Labeled Configuration, to attach the label to the revision with the label you specify.

  The existing view and revision labels are listed in reverse chronological order based on the time for which they were created. The view labels precede the revision labels in the list.

Note Creating a revision label based on another revision label is equivalent to copying that revision label.

- Promotion State Configuration, to attach the label to the revision currently in the promotion state that you specify (the label is attached to the revision that also has the promotion state's current view label).
- Configuration As Of, to attach the label to the revision that was the tip revision at the specified date and time.

To attach an existing label to an item:

1 Select the item from the upper pane.

2 Select Labels > Attach from the item or context menu.

The *Attach a Label* dialog lists all the existing labels and identifies them as view or revision labels. By default, both types of labels are listed because both the View Labels and Revision Labels check boxes are selected.

3 (Optional) You can limit the list to only view labels or only revision labels by clearing the check box for the type of labels you do not want to display.

4 Select a label from the list.

5 Indicate what revision of the item is to receive this label by selecting a configuration option button. Your choices are:

- Current Configuration, to attach the label to the tip revision.
- Labeled Configuration, to attach the label to the revision with the label you specify.

  The existing view and revision labels are listed in reverse chronological order based on the time for which they were created. The view labels precede the revision labels in the list.
- Promotion State Configuration, to attach the label to the revision currently in the promotion state that you specify (the label is attached to the revision that also has the promotion state's current view label).
- Configuration As Of, to attach the label to the revision that was the tip revision at the specified date and time.

To attach a label to a specific revision:

1 Select the item from the upper pane.

2 Select the Label tab.

3 Right-click the revision.

4 Select Attach from the Label context menu. The *Attach a Label* dialog appears.

5 (Optional) You can limit the list to only view labels or only revision labels by clearing the check box for the type of labels you do not want to display.

6 Select a label from the list.

To detach a label from an item:

1 Select the item from the upper pane.

2 Select Labels > Detach from the item or context menu.

The *Detach a Label* dialog lists all the available labels. By default, both types of labels are listed because both the View Labels and Revision Labels check boxes are selected.

3 (Optional) You can limit the list to only view labels or only revision labels by clearing the check box for the type of labels you do not want to display.

4 Select the label to be detached from this item.

To detach a label from a specific revision:

1 Select the item from the upper pane.

2 Select the Label tab.

3 Right-click the label.

4 Select Detach from the Label context menu.

5 Click OK to confirm.

To select the items in the upper pane that have a particular label on some revision:

1 Choose Select > By Label from the item or context menu.

The *Select A Label* dialog lists all the existing labels and identifies them as view or revision labels. By default, both types of labels are listed because both the View Labels and Revision Labels check boxes are selected.

2 (Optional) You can limit the list to only view labels or only revision labels by clearing the check box for the type of labels you do not want to display.

3 Select the label of interest. Items that have this label become selected in the upper pane.

Usually you select items because you want to use them as a group. For example, you can print reports about them, flag them, attach a label to them, or check them out (if they are files).

The status bar indicates the number of items in the upper pane and the number of items selected from the pane (for example, 8 files, 1 selected).

Access Rights that Apply to Labels

Access rights that apply to labels can be set at the view level and at the folder/item levels. The access rights that allow a user or group to create labels, edit their properties, and delete them are set at the view level. For example, if you can create a label, you can set its initial properties. However, if you do not have the right to edit label properties, you cannot later freeze or unfreeze that label.

A label can be attached to individual folders or items, detached from them, or moved from one of their revisions to another. The right to move a label is named “Adjust a label”. These rights can be granted or denied at the folder or item level.

Chapter

17

Using Links

A link is a connection between two folders, two items, or a folder and an item. Creating links can have a number of benefits. For example, when you are reviewing a file and the change requests that affect that file have been linked to it, you can review the change requests without selecting the Change Request tab. This is done by selecting the file, then the Link tab beneath the lower pane, and then double-clicking the appropriate link.

In addition, linking files to change requests enables you to mark the change requests as fixed when you check in the corresponding files. In turn, if you link each set of files to the requirements document that the files fulfill, you can easily refer to or update the document. For more information on linking files to other items, see “Checking In Files” on page 134, “Linking Files to Process Items” on page 213, and “Using the Links to a Process Item” on page 213.

A link does not provide a connection to a single share (or reference), but to all related shares and branches of an item. Links are not affected by any item operations, such as branching, moving, sharing, etc.

Linking Folders and Items

If several items of the same kind (for example, change requests) are going to be linked to the same item (for example, a file), you can create all of the links simultaneously. You select several items from the upper pane, then use the Link button or menu command to connect them to another folder or item. You cannot link folders using the Link button.

To link one folder or item to another:

1. Make your first selection (such as a folder from the folder hierarchy or a file from the upper pane). When selecting from the upper pane, you can select more than one item. When selecting from the folder hierarchy (the right pane), you can select only one folder at a time.
2. Select Links > Create Link from the Folder or an <item> menu or from one of their context menus. Alternatively, if you initially select a file, change request, topic or task, you can use the Link button on the toolbar. This button is disabled if you start the link with a folder.

   The mouse pointer changes to a knotted rope.

3 Make your second selection: one folder or one or more items of the same type.

You may need to change tabs, use All Descendants (either the button on the toolbar or the command on the item menu), and so on to locate your second selection.

4 Select Links > Complete Link from the Folder or <item> menu or from one of their context menus.

You can click the Link button on the toolbar a second time to complete the link to an item. The Link button is disabled if you are linking an item to a folder.

5 To check what you have done, do one of the following:

- Select a linked file, change request, topic or task, then select the Link tab beneath the lower pane to view the links for that item.
- Select a linked folder. Select Folder > Properties, then select the Link tab to view the links for that folder. (The *Folder Properties* dialog does not display the Link tab if you do not have the access rights that allow you to see links.)

You can view a link by selecting either of its ends. The end you select, whether a folder or an item, is called the source. The other end of the link is called the target and is listed in the Item Type column in the Link tab. You can select either end, so which end is the source and which end is the target changes.

By default, a link connects the tip revisions of the linked pair. For example, if you link a file to a change request, the connection floats, always connecting the tip revision of the file to the tip revision of the change request.

You can change the link so that a particular file revision is connected to a particular revision of the change request. The connection becomes fixed (or pinned) rather than floating.

If you change your mind about creating the link after you have started the process but before you have completed it, you can select Links > Cancel Link from the Folder, File, Change Request, Topic, or Task menu or from one of their context menus. If you are using the Link button on the toolbar, press Esc.

In either the link list (in the lower pane) or the Folder Properties dialog, you can right-click the column headers to sort and group the links. For example, to sort the links by the type of item to which they are attached, click the Item Type column header. See “Sorting and Grouping the Data” on page 54 for more instructions.

Understanding the Columns in the Link List

The columns in the link list are listed below along with a description of each. These columns also appear on the Link tab of the *Folder Properties* dialog. (The *Folder Properties* dialog has no Link tab if you do not have the access rights that allow you to see links.)

Each link has two ends: a source and a target. The source is the item selected from the upper pane or from the folder hierarchy (if you are viewing the Link tab of the *Folder Properties* dialog).

Each row in the link list defines a link that has the source item or folder as one of its ends. Because you cannot see the other end of the link, columns in the link list identify it for you.

Created By	The name of the user who created the link.
Created On	The date/time the link was created.
View	The assumption is that you want to locate items in the current view whenever possible, so the view field contains one of the following: The name of the current view if the link was created in the current view, or if the link was created in another view but a shared copy of the item (to which the current item is linked) exists in the current view. The name of the view containing the link. If the view is a reference view, this is the name of the parent view.
Folder	The name of the folder in which the folder or item in the link list resides.
Item Type	Identifies the type of item the target end of the link is attached to. This item is the one listed in the link list. The values that can appear in this column are Folder, File, Change Request, Task, and Topic.
Item	Identifies the item that the target end of the link is attached to by its folder name, file name, change request number, task number, or
Item Details	Describes the item, using a folder description, file description, change request synopsis, task name, or topic title.
Item Version	When the ends of a link are pinned to specific revisions of the folders or items, those revisions appear in the Item Version and the Selection Version columns of the link list (if that revision is in the current view).
Selection Version	The Item Version displays the version number of the target end of the link if that revision is in the current view. The Selection Version displays the revision number the source end of the link. The source is either the folder selected from the folder hierarchy (when viewing a folder's links) or the item selected from the upper pane (when viewing an item's links). When no revision number is in the column, that end of the link is floating rather than pinned.
Comment	From the *Link Properties* dialog, you can enter a comment about this particular link. That comment appears in this column.
Folder Path	Shows folder path information only when linked item is in the same view. Otherwise, displays the following message: "Unavailable. Item in another view."

Selecting Specific Revisions for Links

Each end of a link has an associated start revision and an end revision that determines the range of revisions to which the link applies. The start revision is always fixed at the time of the creation of the link and is set to the first revision on the current branch. The end revision is under the user's control and may be fixed (or pinned), which puts an upper bound on the linked revisions, or floating, which does not. If a link end is pinned, it is always attached to the same version of the linked folder or file. If a link end floats, it moves from revision to revision, as new revisions of the linked folder or item are created.

Determining whether a link is visible on a given item is simple. If any of the revisions between the start and the end revision defined for the link are in the history of the selected item, it is visible. Otherwise, it is not.

To attach link ends to tip revisions:

1. Select a folder or item that you have linked.
2. Do one of the following:
   - If you selected an item, select the Link tab from the lower pane. The Link tab lists the links to other items.

- If you selected a folder, select Folder > Properties, then the Link tab in the *Folder Properties* dialog. (If you do not have the access rights that allow you to see links, there is no Link tab.)

3 From the Link tab, select one or more links.

4 Right-click to open the *Link* context menu.

5 Do one of the following:

- For any link selected in the Link tab, the Pin Link > To Source Item At Tip command pins the link to the tip revision of the link's source. Here source means the folder whose properties you are reviewing or the item selected from the upper pane.

  For example, if a file is selected from the upper pane and the link to a change request is selected from the *Link* tab, this command pins the link between them to the tip revision of the file.

  As another example, if you select a link to a file from a folder's *Folder Properties* dialog, this command pins the link between them to the tip revision of the folder.

- For any link selected in the Link tab, the Pin Link > To Target Items At Tip command pins the link to the link's target. Here target means the folder or item in the Link tab's Item column.

  For example, if a file is selected from the upper pane and a change request is selected from the Link tab, this command pins the link between them to the tip revision of the change request.

- For any link selected in the Link tab, the Pin Link > To Source And Target Items At Tip command pins the link's source and target.

  For example, if a file is selected from the upper pane and a change request is selected from the Link tab, this command pins the link between them to the tip revision of both the file and the change request.

- For any link selected in the Link tab, the Float Link > To Source Item command makes the link's source float from tip revision to tip revision as new revisions are created. Here source means the folder selected from the folder hierarchy or the item selected from the upper pane.

  For example, if a file is selected from the upper pane and a change request is selected from the Link tab, the link between them floats from tip revision to tip revision of the file.

- For any link selected in the Link tab, the Float Link > To Target Items command makes the link's target float from tip revision to tip revision as new revisions are created. Here target means the folder or item in the Link tab's Item column.

  For example, if a file is selected from the upper pane and a change request is selected from the Link tab, the link between them floats from tip revision to tip revision of the change request.

- For any link selected in the Link tab, the Float Link > To Source And Target Items command makes both the link's source and target float from tip revision to tip revision.

  For example, if a file is selected from the file list and a change request is selected from the Link tab, the link between them floats from tip revision to tip revision of the file and from tip revision to tip revision of the change request.

To link specific revisions:

1 Select a folder or item that you have linked.

2 Do one of the following:

- If you selected an item, select the Link tab from the lower pane.

- If you selected a folder, select Folder > Properties, then the Link tab. (If you do not have the access rights that allow you to see links, there is no Link tab.)

3 Select a link from the *Link* tab.

4 Right-click to open the context menu.

5 Select Link Properties. The *Link Properties* dialog appears.

6 (Optional) Enter a link description or comment in the *Comment* text box. The description you enter here appears in the Comment column of the link list.

7 In the *Source Item* group box, do one of the following:

- Click Pin to select a specific revision of the folder selected from the folder hierarchy or of the item selected from the upper pane. The *Select Version* dialog appears.
- Select the folder or item revision from the list. This revision number will appear in the Selection Version column of the link list.
- Click Float so that the link is always connected to the tip revision of this item.

8 In the *Is Linked To Target Item* group box, do one of the following:

- Click Pin, to select a specific revision of the item identified in the link list's Item column. In the resulting *Select Version* dialog, select the folder or item revision from the list. This revision number will appear in the Item Version column of the link list.
- Click Float, so that the link is always connected to the tip revision of this item.

9 Click OK.

Reviewing Link Properties

From the Link tab, you can edit the link's properties, or display the target folder or item's properties in a read-only dialog. Your ability to modify a link is governed by your access rights on the linked items. For operations that affect the link in general (such as Comment), you must have access rights to both items. For operations that affect only one end of the link, rights are required only for the item at that end.

To review the link's properties:

1 Select an item from the upper pane or a folder from the folder hierarchy (right pane).

2 Do one of the following:

- If you selected an item, select the *Link* tab from the lower pane.
- If you selected a folder, select Folder > Properties, then the *Link* tab. (If you do not have the access rights that allow you to see links, there is no Link tab.)

3 Right-click a link from the *Link* tab and select Link Properties from the context menu.

4 In the resulting *Link Properties* dialog, edit these properties where appropriate.

5 Click OK.

To review a linked item's properties or linked folder's properties:

1 Select an item from the upper pane or a folder from the folder hierarchy (right pane).

2 Do one of the following:

- If you selected an item, select the *Link* tab from the lower pane.
- If you selected a folder, select Folder > Properties, then the *Link* tab. (If you do not have the access rights that allow you to see links, there is no Link tab.)

3 Do one of the following:

- Double-click a link from the *Link* tab.
- Right-click a link to open the *Link* context menu and select Item Properties.

The *Properties* dialog displays information about the folder or item that is at the target end of the link. It is the folder or item in the Link tab's Item column. This dialog is read-only.

Selecting a Linked Item

By selecting a single item from the Links pane, you can quickly select its parent, its type, and the item in the list or tree on the upper pane.

To select a linked item :

1. From the Links pane, select an item from the current view.
2. Right click in the Links pane and choose Select Linked Item from the context menu:

   This action highlights the item's parent folder in the folder tree, the item type in the items pane, and the item itself in the upper pane of the main window.

   If the item is not visible with the current filter and query settings, the user sees a warning that the selection could not be completed.

Working with Files Linked to a Specific Item

After selecting a specific file, change request, requirement, task, or topic in the upper pane, you can perform a variety of operations on the files linked to that item. For example, the files can be checked in or checked out. You can also select all linked files or add files to the current selection.

To work with the files linked to a specific item:

1. From the upper pane, select an item in the current view. Alternatively, you can use the selection method described in "Deleting Links" on page 233.
2. Right-click in the upper pane, and select Linked Files from the context menu
3. Highlight one of the following options:
   - Check In: When this option is selected, the system presents the standard *Check In* dialog so that you can check the linked files in.
   - Check Out: When this option is selected, the system presents the standard *Check Out* dialog (minus the View Configuration option) so that you can check the files out. If the link is pinned, the linked files checked out will be the revisions that were linked. If the link is floating, the linked files checked out will be the tip revisions.
   - Select All: When this option is selected, the window focus changes to the File tab and all linked files in the current view are selected. You can then check the files in or out or perform other operations. If a linked file is not visible with the current filter and query settings, the user will see a warning that the selection could not be completed.
   - Add to Selection: When this option is selected, the window focus changes to the File tab and all of the linked files from the current view are added to the current file selections. You can then check the files in or out or perform other operations. If a linked file is not visible with the current filter and query settings, the user will see a warning that the selection could not be completed.

Working with Files on the Links Pane

When one or more files from the current view have been selected in the Links pane, you can check them in, check them out, attach a label to them or detach a label from them. If a single file has been selected, you can compare its contents to another revision.

To work with files on the Links pane:

1. Select a file from the upper pane or the folder from the folder hierarchy (right pane).
2. Click on the Links tab. This action displays all files that are linked to the file you selected.
3. Select one or more of the files on the Links pane, and right-click to display the context menu.
4. Select one of the following options:
   - Check In: When this option is selected, the system presents the standard Check In dialog and the files will be checked in.
   - Check Out: When this option is selected, the system presents the standard Check Out dialog (minus the View Configuration option). If the link is pinned, the files checked out will be the revisions that were linked. If the link is floating, the files checked out will be the tip revisions.
   - Attach Label: When this option is selected, the system presents the standard Label Selection dialog (minus the View Configuration component). If the link is pinned, the files attached will be the revisions that were linked. If the link is floating, the files attached will be the tip revisions.
   - Detach Label: When this option is selected, the system presents the standard Detach Label dialog. The selected labels will be detached from the selected files.
   - Compare Contents (only one file selected): When this option is selected, the system presents the Visual Diff tool. For a pinned link, the local file on disk is compared to the revision that is linked. For a floating link, the local file is compared to the tip revision. If the files are identical, the system will send a message indicating this fact.
5. Complete the tasks required for the option you selected. After you do so, the operation will be performed on the selected files in the Links pane.
6. If requested, click OK.

Deleting Links

When a link was a mistake or has become obsolete, you can delete it.

To delete a link:

1. Select the item from the upper pane or the folder from the folder hierarchy (right pane).
2. Do one of the following:
   - If you selected an item, select the Link tab from the lower pane.
   - If you selected a folder, select Folder > Properties, then the *Link* tab. (If you do not have the access rights that allow you to see links, there is no Link tab.)
3. Right-click the link to be deleted.
4. Select Delete Link from the context menu.

Chapter

18

Using References

A folder or item in one application location can be based on another folder or item stored in a different location within the same server configuration. References indicate the relationships between an original folder or item and the others based on it. References can be used to decide whether the changes you have made to a folder or item in one location need to be applied elsewhere.

At least one reference to a folder or item is created whenever:

- A folder or item is created or added to the application.
- A branching child view is created that will contain that folder or item.

  As a branching child view is created from its parent, a subset of the parent's folders or items becomes part of the child view. The folders and items in that subset are automatically shared into the child view.
- A folder or item is manually shared from one location to another.

As a folder or item is added, shared, or moved, more than one reference to it may occur if the view is a child view that branches and floats, or if the view has child views that branch and float.

For example, suppose you want to move a file from one folder to another in the same view. Suppose that the view has two child views, both of which contain the file. That means that there are at least three references to this file, one in each of three views. Now, you move the file to another folder in the same view. The reference in the current view is moved to represent the file's new location. Depending on the properties of the two child views, a new reference may be created for the file in each of the child views. The references in those child views to the file in its original location still exist, because the application does not assume that you want to change those references just because you have moved the file in the current view. You may end up with five references to this file that formerly had three references.

Note Most administrators avoid branching, floating views if users are likely to perform many operations that result in additional references. For example, moving and sharing can result in multiple unwanted references to the same folders or items, which can cause confusion.

The following table explains what references are created in the current view, the recipient view, the recipient view's parent, and the recipient view's children. This is often recursive. For example, if a reference is created in the parent view, new

references might be created in that view's other children or in that view's parent, and so on, depending on what views are floating.

Table 18.1 What Causes References to be Created?

When a folder or item is:	Is a reference added to the recipient's view?	Is a reference added to the recipient view's parent view?	Is a reference added to the recipient view's child views?
Part of a newly created view	Yes, unless the new view is a reference view. In this case, a new view is not really being created, because a reference view is just a new way of looking at an existing view.) There is one reference for the folder or item in the newly created view.	No, because the parent view is the source of the folder or item, so the reference in the parent view already exists.	No, because the newly created view has no child views.
Added to current view	Yes, there is one reference for the new folder or item in the current view.	Yes, if the current view is a branch none, floating child of the parent view. Otherwise, no.	Yes, if the child view is a branching (either branch none or branch all), floating child of the current view. Otherwise, no.
Shared within current view	Yes, a new reference is created for the shared folder or item in the new location in the current view.	Yes, if the current view is a branch none, floating child of the parent view. Otherwise, no.	Yes, if the child view is a branching (either branch none or branch all), floating child of the current view. Otherwise, no.
Moved within current view	No, the original reference is updated to reflect the move.	Yes, if the current view is a branch none, floating child of the parent view. Otherwise, no.	Yes, if the child view is a branching (either branch none or branch all), floating child of the current view. Otherwise, no.

Viewing References

Because of manual sharing and because views are children of other views, a folder or item can be associated with more than one project, view, or parent folder (within the same server configuration). Each instance of the folder or item has a reference to its tip revision.

To view folder references, you open a separate dialog; to view item references, you use the project view window and the Reference tab on the lower pane.

To view folder references:

1. Select the folder from the folder hierarchy.
2. Select Advanced > References from the *Folder* or context menu.

   The resulting *Folder References* dialog displays a tree that indicates which project views reference this folder, and their relationship to each other.

In the following sample *Folder References* dialog, the selected folder has four references.

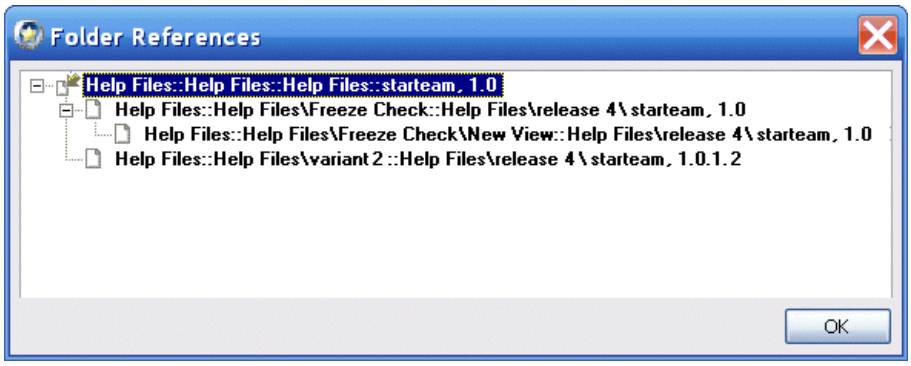
The Current icon indicates which reference represents the currently selected folder or item. Otherwise, this dialog contains the same information regardless of the view in which you selected the folder.

Each reference shows the following, separated by double colons (::):

- The project name (for example, Help Files).
- The path from the root view to the view containing the folder (or item). For example, Help Files\Freeze Check\New View, where Help Files is the name of the root view, Freeze Check is a child of the root view, and New View is a child of the Freeze Check view.
- The path to the folder within the view. In the case of an item, the path is to the item's parent folder.
- In the case of an item, the name or number associated with that item. This can be the file name, change request number, the requirement number, the task number, or topic number.
- The tip revision number for the folder (or item) in that view. (This information is separated from the rest of the reference by a comma, rather than the double colon.)

  For example, the folder in the figure is revision 1.0 in all views except for the variant 2 view (see the last leaf in the tree). In the variant 2 view, the folder's revision is 1.0.1.2 which indicates that the folder has been branched from the 1.0 revision in its parent view and has had three revisions in the variant 2 view. Those revisions are 1.0.1.0, 1.0.1.1, and 1.0.1.2.

In this example, a project's name, its root view's name, and the root folder in the root view all have the same name.

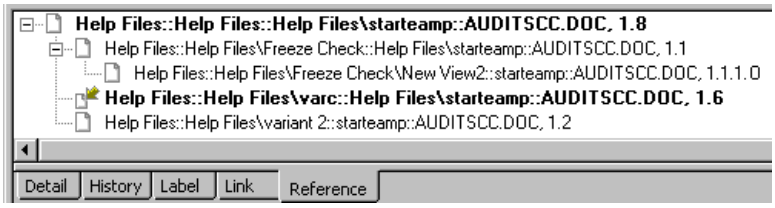


This dialog can be resized (by dragging an edge or corner). It displays scroll bars when appropriate.

The references in bold indicate which revisions of the currently selected folder or item are its descendants. In other words, the currently selected folder or item is part of the revision history for the references that are in bold.

Consider the following figure that shows the references for a file (auditscc.doc). The reference for the currently selected file indicates that file's revision is 1.6. As indicated by the bolding of its reference, revision 1.8 is the only descendant of revision 1.6. If a defect is found in revision 1.6 of auditscc.doc, the bolding helps you determine which

descendants of 1.6 may also need the corrected lines. In this case, only 1.8 may need to be updated.



To view item references:

1 Select the item from the upper pane.

2 Click the Reference tab beneath the lower pane.

Data similar to that in the *Folder References* dialog displays in the lower pane. The *Reference* pane has no context menu.

To view references for past revisions of an item:

1 Display the history list by doing the following:

   a Select an item from the upper pane.

   b Select the History tab below the view window's lower pane.

2 Select and right-click the revision in the history list. The *History* context menu opens.

3 Click References.

An appropriate *References* dialog appears. It contains data similar to that in the *Folder References* dialog. The references are the same for a historical or tip revision. However, bolding will vary.

To view references for past revisions of a folder:

1 Select the folder from the folder hierarchy.

2 Select Properties from the *Folder* or context menu.

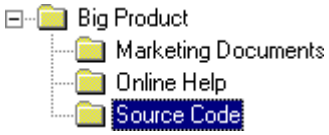
3 Select the History tab. (There is no History tab if you do not have the access rights that allow you to see a folder's history.)

4 Select and right-click the revision in the history list. The *History* context menu opens.

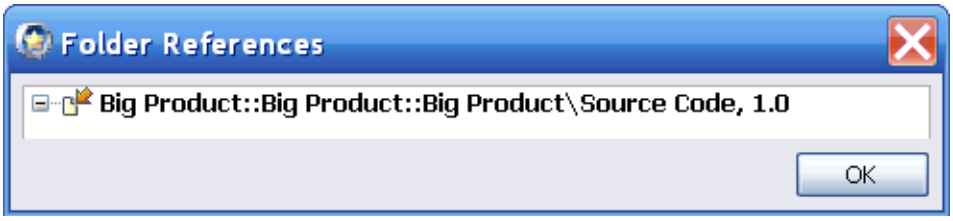
5 Click References. An appropriate *References* dialog appears.

Understanding the Initial Reference

When a folder or item is added to the application, a reference is created. The following figure shows a folder hierarchy for a newly created project. (In this case, it is the folder hierarchy for the root view of that project.)



The next figure shows the *Folder References* dialog for the Source Code folder in this view. At this point in time, there is only one reference for this folder.



A reference has several parts, including the project name, the view path, the folder path, and the tip revision number.

As changes are made to this folder's properties, the revision number might change from 1.0 to 1.1 and later 1.2. However, there still will be only one reference to this folder.

If a reference view is created (to be used, for example, by a group of reviewers), the view hierarchy for the Big Product project would contain two views, but the Source Code folder would continue to have just one reference.

A Reference view contains a subset of the folders in its parent view, but those folders are the same folders as those in the parent view. They cannot branch. See the *StarTeam Administator's Guide* for more information about reference views.

Do not confuse reference views with folder and item references. A reference view looks like a new view, but it is really a subset of an existing view. A folder or item reference is like a reference count. It indicates how many copies of the object exist or can exist if the object branches in each of its new locations.

The irony of using the word reference in both of these terms is that the creation of a reference view does not result in the creation of any folder or item references. The following figure shows the *Folder References* dialog after the creation of the reference view for reviewers.

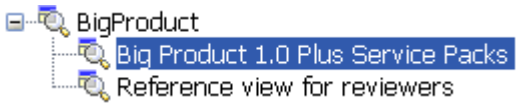
It is not until the folder is shared manually by a user, or shared automatically when a branching child view is created, that additional references appear in the *Folder References* dialog.

Understanding References Created by Branching Views

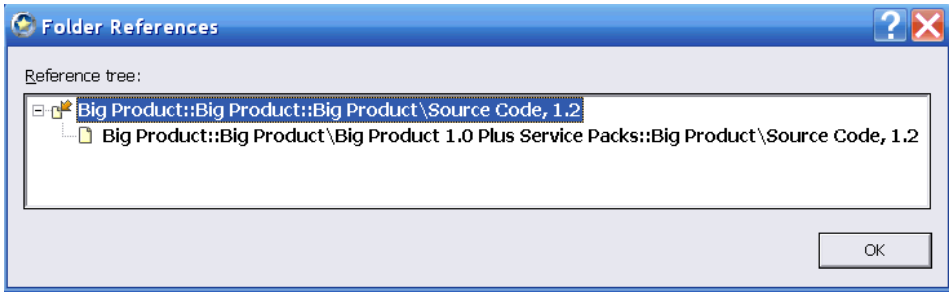
When a branching view is created, each folder or item automatically shared from the parent view to the child view acquires an additional reference. In the reference hierarchy, the new reference is a child of the original reference.

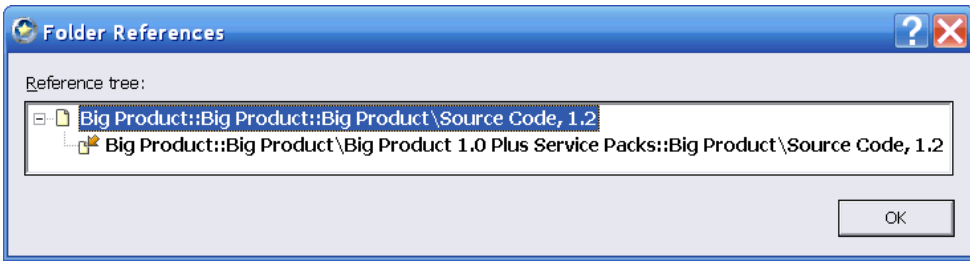
Suppose that when the 1.0 version of Big Product ships, the team leader creates a branching view (based on the ship date for the 1.0 version) to be used for service

packs, while new development on version 2.0 still continues in the project root view. The following figure shows the folder hierarchy for all three views.



At this point the reference count goes to two. The following figures show the *Folder References* dialog for the Source Code folder in the root view and in the child view created for 1.0 service packs.



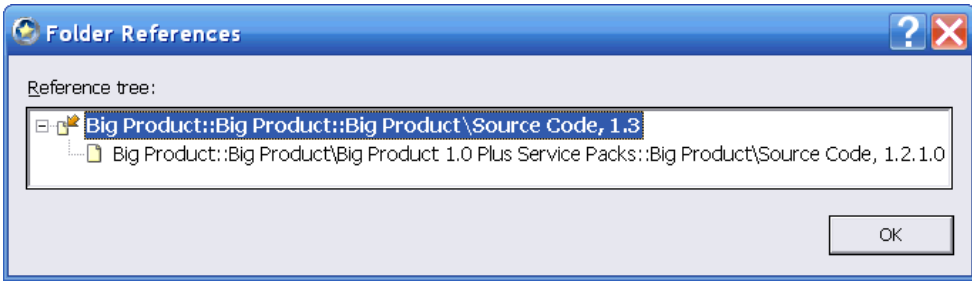


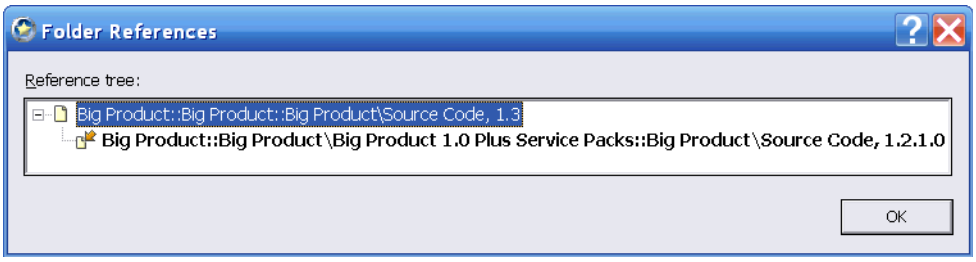
The Current (You Are Here) icon indicates which reference represents the currently selected folder or item. Otherwise, this dialog contains the same information regardless of the view in which you selected the folder.

The reference for a child view is indented beneath the reference for its parent.

The references in bold indicate which revisions of the folder or item are descendants of the folder or item with the Current (You Are Here) icon. In other words, the current folder or item is part of the history for the revisions that are in bold.

In the previous two figures, both references were bold. In the next two figures, this is not the case. The properties of the Source Code folders in both the parent view and the child view have changed. The parent's folder now has revision 1.3, and the child's folder now has the revision 1.2.1.0. Both folder's histories have taken off in new directions.

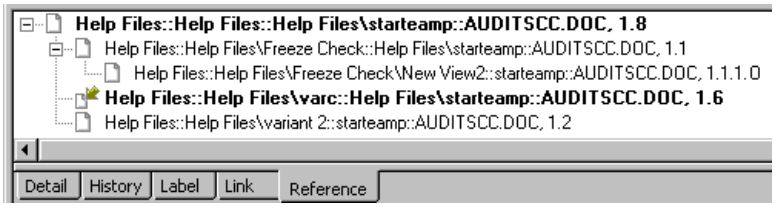




The current folder is considered a descendant of itself, so it is always bold. However, now the folder in the other view has evolved. It is no longer in the history of the current folder.

When you look at the history of a folder or item, you see its ancestors, not its descendants. However, if you change the tip revision in one location and that revision is an ancestor of the tip revision in another location, you might want to apply your change to the tip revision in the other location (the first object's descendant) as well. The way to tell if a revision has descendants is to look at its references.

Consider the following figure that shows the references for a file (auditscc.doc). As the boldface type indicates, if the current revision is 1.6, then 1.8 is its only descendant. This also means that revision 1.6 would be found in the history for 1.8.



If a defect is found in revision 1.6 of auditscc.doc, the bolding helps you determine the descendants of 1.6 that may also need the corrected lines. In this case, 1.8 may need to be updated. The other references are for revisions of the file that:

- Have already diverged (branched) and may be quite different than the current file.
- Are ancestors of the current file and less likely to need a change. For example, they may be in views that are read-only or no longer in use. Whatever the reason for the gap, the ancestors might require far more work that the just the changes you are about to check in.

You should check for descendants before (and perhaps after) you create a new revision of a folder or item. Before the change becomes a new revision in the application, you can see the descendants. Afterwards, you may see what other references have the same revision number as the newly changed folder or item. If they, too, have the new revision number, then they, too, already have the new change. For example, the file may be floating in other views.

Understanding References Created by Adding Items to Views

The addition of a new folder or item to a parent or child view can result in one or two references, depending on the relationship between the two views.

If the child view is a branching, floating view, a reference is created in each view when a new folder or item is added to the parent.

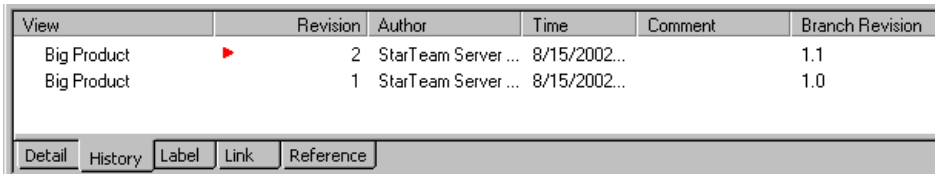
If the child view is a branching, floating view created using the Branch None option, a reference is created in each view when a new folder or item is added to the child.

Floating Downwards

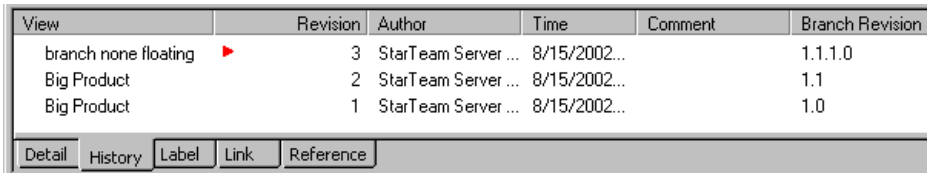
When a view has a branching child view (whether created with the Branch None or Branch All option) and the child view is floating, any folder or item added to the parent view becomes visible in both views. The history of the folder or item indicates the view in which the object was created, and the reference hierarchy displays the reference that identifies the parent view as the parent reference.

For example, if a file is added to the parent view, its history in either view shows the name of the parent view—until the file branches in the child view.

The following figure shows the history in the parent view for a file that was added to the parent view and floated downwards.



The following figure shows the history in the child view for a file that was added to the parent view and floated downwards. The file's history displays the name of the view from which the file was originally added to the application—until the file branches. Then it displays the name of the view in which the file branched.



The next figure shows the references for this file after it has branched in the child view.

Notice that the history clearly shows the parent view as "Big Product" before the file branches. The history and references for folders and items added to the parent view are similar to those for folders and items that were in the parent view at the time the child view was created.

Note The name of the views in these figures makes the figure easier to understand. You would probably never name a view "parent view" or any other of the names shown in the figures.

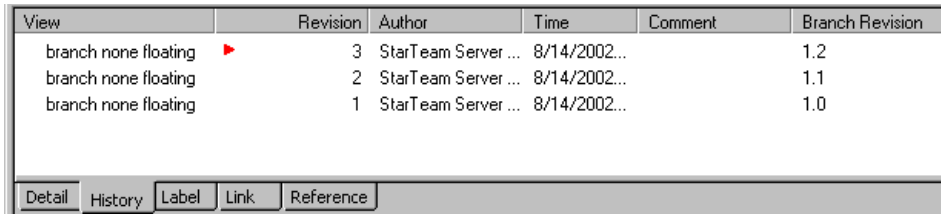
Floating Upwards

When a view has a branching child view (created with the Branch None option) and the child view is floating, any folder or item added to the child view becomes visible in both views. This is not true of branching, floating child views that were created using the Branch All option.

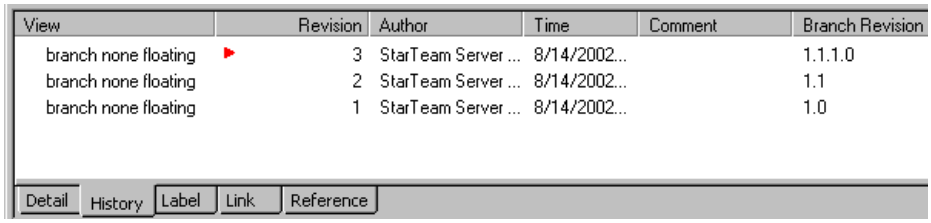
The history of the folder or item indicates the view in which the object was created, but the reference hierarchy always displays the reference that identifies the parent view as the parent reference.

The following figure shows the history in the parent view for a file that was added to a child view and floated upwards. Notice that, even though this is the history in the parent

view, the history displays the name of the view from which the file was originally added to the application.



The following figure shows the history in the child view for a file that was added to the child view and floated upwards. The file's history displays the name of the view from which the file was originally added to the application—until the file branches. Then it displays the name of the view in which the file branched. In this case, those two views just happen to be the same view.



The following figure shows the reference hierarchy for a file that floated upwards. Notice that from the references, you cannot tell that the file was added to the application from the branching child view.

The file's history enables you to determine where the file originated.

Floating Upwards and Downwards

If the view hierarchy is deep (the root view has grandchildren, great-grandchildren, and so on), the use of branching, floating views can cause a great deal of confusion. For example, suppose you add a file to a grandchild of the root view. Further suppose that this grandchild view was created using the Branch None option and that its parent (a child of the root view) was created using the Branch None option. The file you add can float up to the parent and grandparent of the current view from which it will, in turn, float back down to the current view. This results in:

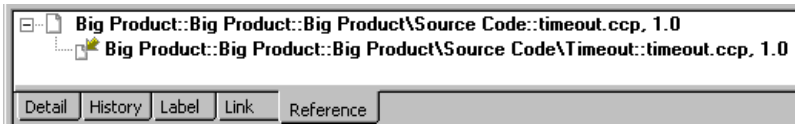
- One reference to the file in the current view
- One reference to the file in the current view's parent (the result of floating up from the current view)
- One reference to the file in the root view (the result of floating up from the current view's parent)

More references are created if the current view has floating children, grandchildren, and so on. Still more are created if the root view or parent view have other floating children besides the ones mentioned above.

Understanding References Created by Manually Sharing Objects

As you share a folder or item from one location to another (whether in the same view or a different one) an additional reference is created for that object in the new location. The reference for the new folder or item becomes a child of the reference from the folder or item that was shared.

The following figure shows two references for a file named timeout.cpp. The file was manually shared from a folder named Source Code to a folder named Timeout in the same view. Notice that these references are very similar to those shown earlier for two folders, the second based on the first, but created as a by-product of creating a branching view.



The application does not differentiate between references based on what caused them to be created. However, you can tell from the hierarchy that the first reference is the source of the second reference, because the second reference is indented under the first. You can also tell, because they are in the same view, that a manual share or move occurred. (The second reference would be in a different view if it was created automatically when a child view was created.

A shared folder or item can branch, but may never do so. Regardless, some subset of its history is part of the history of the original folder or item.

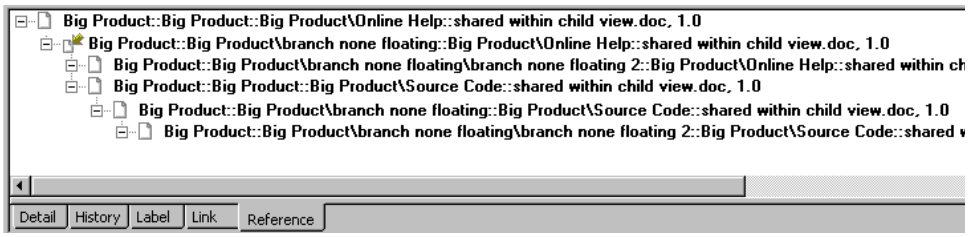
Floating Upwards and Downwards

If the view hierarchy is deep (the root view has grandchildren, great-grandchildren, and so on), the use of branching, floating views can cause a great deal of confusion. Suppose all the views except the root view branch and float. At its new location, depending on how views were created, the folder or item you share can float:

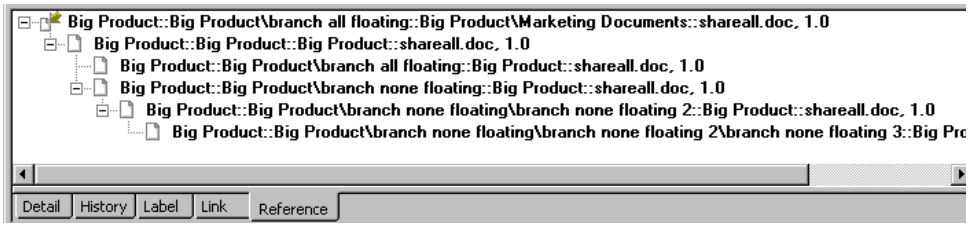
- Up the view hierarchy from the recipient view to the root view
- Down to all the recipient view's children, grandchildren, and so on
- From the recipient view's parents, grandparents, and so on, to all of their other children

This can result in a reference to the folder or item in the new location in every view in the project's view hierarchy. Many of those views may have already had a reference to the folder or item in its old location.

The following figure shows all the references created by sharing a file named "shared within child view.doc" from one location in the "branch none floating" view to another location in that same view. The first three references are the references that existed prior to the sharing operation. The fourth reference is the new reference in the root folder. It is shown as a child of the first location in the "branch none floating" view because it floated up from that view. The fifth and sixth references resulted from references that floated down to the "branch none floating" view's child view.



The next figure shows that the file named shareall.doc existed only in the “branch all floating” view before it was shared to another view. The reference to the root folder starts the references that occurred as a result of the share operation. However, the recipient view could have been any of the other views, because the file would float up to the root and back down. On the way down, a second reference was created in the “branch all floating” view.

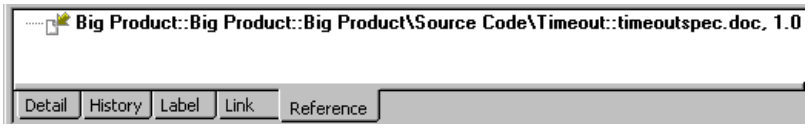


Understanding References Created by Moving Objects

When a folder or item is moved from one location to another within the same view, the object is deleted at the old location and reinstated at the new location. However, there can be side effects in that view’s parents and children if any of the views are floating. This is because the copy at the old location is not deleted except in the current view. The parent and child views may end up with two references (one to the old location and one to the new location) instead of one to the new location.

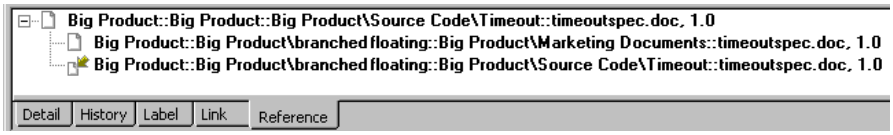
Suppose you move the file named timeout.doc from the Marketing Documentation folder to the Timeout folder in a given view that has no branching child views.

The following two figures show the references for this file before and after the move. The number of references is the same; only the path to the file has changed. The file has been deleted from its original location and added to its new location.



However, suppose this view has a child view that was created without cutting off the connection to the parent (in other words the child view is branched and floating). In the child view, if the moved file has not yet branched, it is not deleted from its old location because you might really still want it here. However, it is added to the new location because it is perceived as a change to the parent that should be reflected in the child.

Notice that the file has only one reference in the parent but that it has two in the child view.



Some customers sort items using folders. For example, they decide to create a series of folders in a view to classify change requests by criteria such as:

- Will definitely make the next release
- Are under consideration for the next release (time permitting)

These change request are usually moved from the root folder to one of the sorting folders, or later rearranged and moved from one sorting folder to another. This is a convenience in the current view, but it can cause multiple references in a parent or child view. If the view hierarchy is deep, the current view's parents, grandparents, children, grandchildren, and so on may be affected. Customers who use such systems usually create child views that do not float.

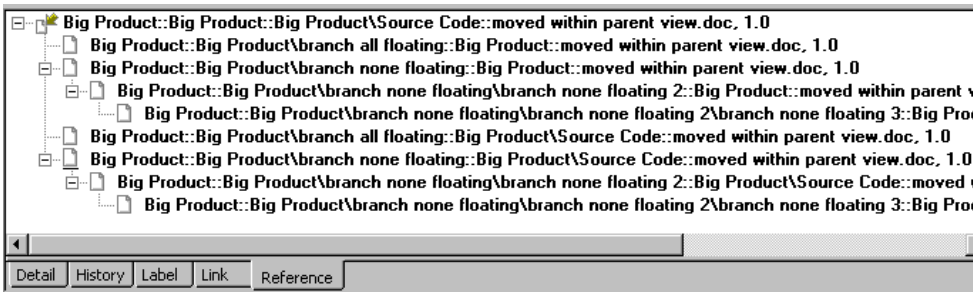
Floating Upwards and Downwards

If the view hierarchy is deep (the root view has grandchildren, great-grandchildren and so on), the use of branching, floating views can cause a great deal of confusion. Suppose all the views except the root view branch and float. At its new location, the folder or item you move can float:

- Up the view hierarchy from the recipient view to the root view
- Down to all the recipient view's children, grandchildren, and so on
- From the recipient view's parents, grandparents, and so on, to all of their other children

A move operation results in one fewer reference to the moved folder or item in the view from which it was moved, and one more reference to it in the recipient view.

The following example shows that a file named "move within parent view" was moved from one location in the root view to another location in that same view (which is why there is only one reference to it in that view). Originally, the file was referenced in five views. The move caused a new reference in all the child views of the root folder, giving each of them two references to the moved file (one reference in its original location and one in its new location).



Chapter 19

Reporting and Exporting Data

The application offers a wide variety of reports, which are pre-formatted in HTML. However, you can modify the HTML formatting or the columns used in the report by modifying the correct template for the report.

Report generation is affected by sorting, grouping, and selecting items in the file, change request, topic, or task list of the view window. Before you generate a report, arrange the data in the view window.

Sample Scenarios for Generating Reports

Reports are generated for a variety of reasons. The following examples describe three possible scenarios for generating a report using change requests and a brief walk-through on how to generate them. For more in-depth instructions, see “Creating Reports” on page 250.

A manager wants a report of all the change requests that have been resolved across the entire project:

1. The manager selects the root folder from the folder hierarchy.
2. Next, the Change Requests tab is selected. This opens the list of change requests in the upper pane.
3. The Status=Resolved filter is then selected from the Status drop-down list box.
4. Next, the All Descendants (either from the toolbar or the Change Request menu) button is activated.
5. Finally, he or she selects Change Request > Reports from the client’s menu bar to generate a Change Request Summary Report.

A team leader prefers to see change requests based on who is responsible for fixing them:

1. The team leader selects the project folder that corresponds to his or her team from the folder hierarchy.
2. Next, the Change Requests tab is selected. This opens the list of change requests in the upper pane.
3. The <Show All> filter is then selected from the Status drop-down list box.

4 Next, the All Descendants button is activated.

5 The team leader then clicks the header of the Responsibility column to sort the change requests based on the responsible team member.

6 Finally, the team leader selects Change Request > Reports from the client's menu bar to generate a Change Request Detail Report.

Individual programmers want to see only those change requests for which they are responsible:

1 The individual programmers select their project folders from the folder hierarchy.

2 Next, the Change Requests tab is selected. This opens the list of change requests in the upper pane.

3 A query is then applied (Responsibility Equals <user name>) to see only the change requests that these individual programmers are currently responsible for.

4 Next, the All Descendants button is activated.

5 Finally, they select Change Request > Reports from the client's menu bar to generate a Change Request Detail Report.

Although these examples apply only to change requests, you can create additional reports for other items. You can also export data for use in spreadsheets and other applications.

Creating Reports

The following section describes how to generate reports in HTML.

To create reports in HTML:

1 Select Reports from the File, Change Request, Topic, Task, or Audit menu.

The resulting *Reports* dialog displays a list of possible reports in the Available Reports list box.

2 From the Available Reports list box, select any type of report.

3 Do one of the following:

- Select the Current Selection option button (in the Create Report For group box), to create the report using only the items selected from the upper pane.
- Select the Select All option button, to use all the items displayed in the upper pane.

4 In the Output File Name text box, enter or browse for the report's name and location—if you do not want to use the default. Be sure to use .htm or .html as the extension.

The default folder for reports is set as a personal option. For more information, see "Customizing the Application" on page 273.

5 (Optional) Enter a name for your report in the Report title text box.

6 Do one of the following:

- Click Generate to view the report. Your web browser opens and displays your report.
- Click Print to print the report as it would appear in the browser. The Print command works only if your computer has a "print" action file association for the .htm extension. Installing a browser on your computer creates that association.

  To print a Detail Report, select Print from the File, Change Request, Topic, Task, or Audit menu.

Exporting Data

You can export data for use in spreadsheets and other applications.

The maximum number of columns that can be displayed in the upper pane is 60. This is also the maximum number of columns you can export—even if the database has more columns than that. Make sure the application that uses the exported data can accept up to 60 columns.

To create text files that can be imported into Excel or other such applications:

1 Do one of the following:

- Select Advanced > Export from the File, Change Request, Topic, or Task menu.
- Select Export from the Audit menu.

The *Export* dialog appears.

The *Export* dialog displays two lists. The Available Fields list box contains all the fields that could be displayed as column headers—but that are not currently displayed. The Show These Fields In This Order list box displays all the fields that are currently displayed.

For a list of the fields available for each item, see the tables starting with "Changing the Fields Specified in the Templates" on page 257.

2 Do any combination of the following:

- Add fields to the report.
  - Select the fields to appear in the report from the Available Fields list. You can display a maximum of 60 fields.
  - Click Add.
- Remove fields from the report.
  - Select the fields to be removed from the Show These Fields In This Order list box.
  - Click Remove.
- Change the order of the fields to be displayed in the report.
  - Drag each field name to the desired location in the Show These Fields In This Order list box.

    Remember that you can double-click a field name to move it from one list box to the other.

The *Show Fields* dialog initially displays the most commonly used fields. Select the Show Advanced Fields check box to select from a complete list of the available fields.

3 Select the Current Selection option button in the Export For group box to create the report using only the items selected from the upper pane. Select the Select All option button to use all the items displayed in the upper pane.

4 Select the Export Field Names check box to include the column header fields you selected in the report. Clear the check box to exclude the column header fields in the report.

5 Select the Replace Embedded Carriage Returns With Spaces check box if you want to remove from the exported data any carriage returns that may be embedded in text fields.

6 Enter a qualifier in the Text Qualifier text box.

The qualifier will be used at both ends of the text fields. This prevents the special treatment for carriage return/line feeds and special characters that occur in HTML output.

Wherever the qualifier appears in the text, it will be doubled. For example, suppose the qualifier is the double quotation mark and the Description column of a Change Request report contains:

```
Type "whatever" then press Enter.
```

The report file will contain:

```
"Type ""whatever"" then press Enter."
```

7 Select a character to separate the data from the Delimiter drop-down list box.

8 Select an encoding from the Encoding drop-down list box. See "Available Encodings" below for a complete list of encodings.

9 In the Output File Name text box, enter or browse for the report folder and location.

Be sure to use .txt or another appropriate extension. If you use no extension, the file will have a .txt extension.

10 Click OK to generate the report.

11 Open the report in the application you selected and follow the print instructions to print the report.

Available Encodings

If you use the Cross-Platform client to export data, you will see some subset of the following as possible encodings for the exported data. The list of encodings depends on what is available on your workstation.

ASCII
8859_1 // ISO Latin 1
8859_2 // ISO Latin 2 (Eastern Europe)
8859_3 // ISO Latin 3 (Latin, Esperanto)
8859_4 // ISO Latin 4 (Baltic - old)
8859_5 // ISO Cyrillic
8859_6 // ISO Arabic
8859_7 // ISO Greek
8859_8 // ISO Hebrew
8859_9 // ISO Latin 5
Big5 // Windows Taiwan
Cp297 // EBCDIC France; cp stands for code page;
Cp278 // EBCDIC Finland, Sweden
Cp936 // Chinese (simplified)
Cp950 // Chinese (traditional)
Cp932 // Japan
Cp1250 // Windows Latin 2
Cp1251 // Windows Cyrillic
Cp1252 // Windows Latin 1; American English
Cp1253 // Windows Greek
Cp1254 // Windows Latin 5 (Turkey)
Cp1255 // Windows Hebrew; Israel
Cp1256 // Windows Arabic
Cp1257 // Windows Latin 4 (Baltic)
Cp1258 // Windows Vietnamese
EUC_CN // cp1383; EUC Chinese
EUC_JP // EUC Japanese
EUC_KR // EUC Korea
EUC_TW // EUC Taiwanese

GBK // Chinese
ISO2022CN // Chinese and Taiwanese
ISO2022CN_CNS // Chinese
ISO2022CN_GB // Chinese
ISO2022JP // Japanese
ISO2022KR // Korean (South Korea)
KOI8_R // Russian/Cyrillic
SJIS // Japanese
UTF8 // UTF-8 Unicode
Unicode

Available Reports

You can view and print a number of different reports, depending on the selected tab.

Table 19.1 Available Reports: Files Tab

Name of Report	Description of Report
Default	Lists information located in the Detail pane, using one line for each field.
Detail	Lists files and their revision histories.
Detail with Description	Lists files, their descriptions and revision histories.
Grouping Summary	Indicates the number of files in each group plus the total number of files.
Links	Lists the selected file names and the items linked to them.
Summary	Lists each selected file (or all of them when none are selected). The report gives the File Name, Status, Locked By, and Revision fields and the total number of selected files.
Summary with Description	Lists each selected file (or all of them when none are selected). The report gives the File Name, Status, Locked By, Revision, and Description fields and the total number of selected files.

Table 19.2 Available Reports: Change Requests

Name of Report	Description of Report
Default	Lists information located in the Detail pane, using one line for each field.
Detail	Lists specific change request fields: Modified Time, Responsibility, Entered On, Entered By, Last Build Tested, Status, Severity, Addressed In, Addressed By, Priority, Test Command, Type, Synopsis, Description, Work Around, and Fix.
Grouping Summary	Indicates the number of change requests in each group plus the total number of change requests.
History	Lists the change request fields found in a Detail report followed by information about each revision of the change request: the revision number, its date and time, author, view, comment, and branch revision
Links	Lists the selected change requests and the items linked to them.
Summary	Lists each selected change request (or all of them when none are selected). The report gives the Modified Time, Responsibility, Entered On, Entered By, Priority, Type, Status, Severity, and Synopsis fields and the total number of selected change requests

Table 19.3 Available Reports: Requirements

Name of Report	Description of Report
Default	Lists information found in the Detail pane (when in list format), using one line for each field. You can create this report while the pane is in either tree or list format.
Detail	Lists specific requirement fields: Modified Time, Name, Type, Status, Owner, Priority, and Description
Grouping Summary	Indicates the number of requirements in each group plus the total number of requirements.
Links	Lists the selected requirements and any items linked to them.
Summary	Lists each selected requirement (or all of them when none are selected). The report gives the Title, Created By, and Created Time fields and the total number of selected requirements.

Table 19.4 Available Reports: Tasks

Name of Report	Description of Report
Detail	Lists specific task fields: Modified Time, Responsibility, Created Time, Created By, Name, Status, Priority, Milestone, Duration, Percent Complete, Needs Attention, Attention Notes, Planned Start, Planned Finish, Planned Work, Actual Start, Actual Finish, Actual Work, and Notes.
Grouping Summary	Indicates the number of tasks in each group plus the total number of tasks.
Links	Lists the selected tasks and any items linked to them.
Summary	Lists each selected task (or all of them when none are selected). The report gives the Task Name, Created By, and Created Time fields and the total number of selected tasks.

Table 19.5 Available Reports: Topics

Name of Report	Description of Report
Default	Lists information found in the Detail pane (when in list format), using one line for each field. You can create this report while the pane is in either tree or list format.
Detail	Lists specific topic fields: Created By, Status, Created Time, Priority, Title, and Description.
Grouping Summary	Indicates the number of topics in each group plus the total number of topics.
Links	Lists the selected topics and any items linked to them.
Summary	Lists each selected topic (or all of them when none are selected). The report gives the Title, Created By, and Created Time fields and the total number of selected topics.

Table 19.6 Available Reports: Audit Entries

Name of Report	Description of Report
Default	Lists information found in the Detail pane, using one line for each field.

Table 19.6 Available Reports: Audit Entries (continued) (continued)

Name of Report	Description of Report
Grouping Summary	Indicates the number of audit entries in each group plus the total number of audit entries.
Summary	Lists each selected entry (or all of them when none are selected). The report gives the Event, Class, Created Time, User, and Item fields and the total number of selected entries.

Using Report Templates

The application allows you to customize report templates. The templates are located in the folder you designated during installation. If you used the default path, the Reports folder is *installation_folder*\Reports. Be aware that different clients and different releases of just one client will probably have different installation folders. You may need to put templates in more than one location. All the templates are in HTML format.

You can open and edit the report templates in any text editor, Microsoft's Developer Studio, or HTML tool. A simple and easy method of creating and editing templates is to use Microsoft Word 97 or later, which includes automated HTML file generation.

The Reports folder includes a series of templates for each type of report. Each template provides the formatting information needed to create a part of the report. For example, the Change Request Default report uses the following templates:

- ChangeDefault.Title
- ChangeDefault.GrpInfo
- ChangeDefault.Group1
- ChangeDefault.EndReport

The *x*Default.Title is processed first and only once. It uses the title you provide in the *Reports* dialog as the Report Title.

The *x*Default.GrpInfo is processed once for each group. Although you can sort data by clicking column headers, this does not result in groups for a report. You must use the Sort and Group feature to arrange groups.

The *x*Default.Group1 template is repeated for each item, in this case, each change request, in the report. It creates a record for the item and separates it from the records for other items with a horizontal line.

The *x*Default.EndReport is processed only once. It totals the number of items in the report.

The following illustration is a sample Change Request Default report. It shows the following:

- Title (formatted by ChangeDefault.Title)
- Two group lines (with gray backgrounds and formatted by ChangeDefault.GrpInfo
- Two records (each formatted by ChangeDefault.Group1)

The File Grouping Summary report uses:

- FileGrouping Summary.Title
- FileGrouping Summary.GroupSummary
- FileGrouping Summary.EndReport

Each group in the sample report contains only one record. However, most reports would have several records in each group. The sample does not show the total (formatted by ChangeDefault.EndReport), but the last line would have been: Total: 4.

Change Request Default Report, Grouped by Status and Priority

Status: New (1 items), Priority: No (1 items)

Change Number: 1
Synopsis: Uninstall does not remove all files, folders and reg keys.
Type: Defect
Status: Fixed
Severity: Low
Addressed In:
Addressed By: Tu
Last Build Tested: 3.00.176
Responsibility: John
Work Around:
Modified On: 4/30/98 3:32:04 PM
Priority: No
Description: Running Uninstall will not remove entries that are not created during its installation.

Test Command:
Fix:
Created On: 12/31/97 4:33:46 PM
Created By: John

Status: In Progress (1 items), Priority: No (1 items)

Change Number: 2
Synopsis: Last used project(s) do not reopen on next use. (need workstation option)
Type: Defect

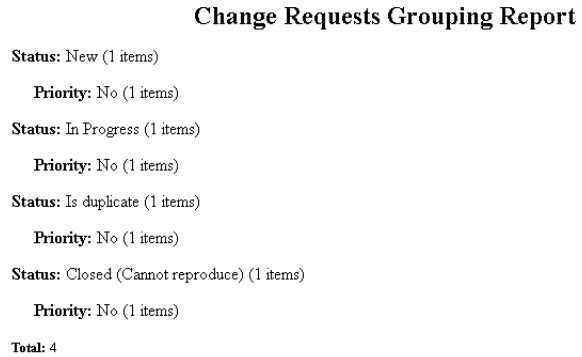
The File Grouping Summary report uses:

FileGrouping Summary.Title

FileGrouping Summary.GroupSummary

FileGrouping Summary.EndReport.

The xGrouping Summary.GroupSummary template is used once for each group. The following figure provides a sample:



Change Requests Grouping Report

Status: New (1 items)

Priority: No (1 items)

Status: In Progress (1 items)

Priority: No (1 items)

Status: Is duplicate (1 items)

Priority: No (1 items)

Status: Closed (Cannot reproduce) (1 items)

Priority: No (1 items)

Total: 4

TemplateDescription

.Title: In the .Title templates, the report title is represented with the `~~ReportTitle~~` tag.
In the .Title templates, you might want to add your company's logo, change the font or size, etc.

.GrpInfo: The .GrpInfo templates contain information describing the group, which is represented as the `~~GroupingInfo~~` tag.

.Group1..9: The .Group*x* files are processed in the numerical order in which they are found. The numbers do not have to be contiguous. The application searches for .Group1 through .Group9 as it creates a record for each item in the report.

The names of the fields in the component are individually specified within the double tildes (~~). You must use the correct SQL name. (See "Changing the Fields Specified in the Templates" on page 257 for details.)

The fields' data replaces the SQL name and the double tildes. To appear in the report, all the specified fields must be displayed as columns in the upper pane at the time the report is generated. Otherwise, those fields are skipped.

.GroupSummary: The .GroupSummary templates are used when totalling the items in a group. The group is represented with the `~~GroupingInfo~~` tag.

.EndReport: This template ends your report. It might include the following total tags:

- ~~TotalHistoryCount~~
- ~~TotalRecordCount~~
- ~~TotalLinkCount~~

Note Because a report has several templates and usually only the first has the <HTML> tag, you may need to add an <HTML> tag to the beginning of a template in order to edit it. Only the last template should ever have the </HTML> tag.

Changing the Fields Specified in the Templates

You can change what fields appear in a report by changing the fields specified in the report's template. Fields must be specified in the templates using their internal identifier. This is usually different from their display name (which is used as the column header, the name displayed in a pane, and the name in the report).

For example, the TopicsSummary.Group1 template contains the following lines:

```
<TD WIDTH=450><FONT SIZE="2">~~Title~~</FONT></TD><TD WIDTH=0></TD>
<TD WIDTH=100><FONT SIZE="2">~~CreatedUserID~~</FONT></TD><TD WIDTH=0></TD>
<TD WIDTH=120><FONT SIZE="2">~~CreatedTime~~</FONT></TD><TD WIDTH=0></TD>
```

If you wanted to add the Read Status field to this report, you would need to know that its internal identifier is Read Status. Then you might add the following line to the template:

```
<TD WIDTH=120><FONT SIZE="2">~~Read Status~~</FONT></TD><TD WIDTH=0></TD>
```

Remember that, even if you specify a field in a template, that field is skipped if it does not also appear as a column in the project view window's upper pane. You might want to create filters for specific reports prior to creating the report. For more information about filters, see "Using Filters" on page 61.

See "Understanding the Fields" on page 65 for a description of each field and the name of its internal identifier. Some additional fields for history and links are listed in the table below.

Table 19.7 Additional Fields for File History and Links

File History	
Display Field Name	**Internal Identifier**
Author	HistoryAuthor
Branch Revision	HistoryBranchRevision
Comment	HistoryReason
Revision	HistoryRevision
Time	HistoryDate&Time
View	HistoryView
Links	

Table 19.7 Additional Fields for File History and Links

File History	
Display Field Item	**Internal Identifier**
Item Type	LinkType
Comment	LinkComment
Created By	LinkCreatedBy
Created On	LinkCreatedTime
Folder	LinkFolder
Folder Path	LinkFolderPath
Item	LinkItem
Item Details	LinkItemDetails
Item Version	LinkItemVersion
Selection Version	LinkSelectionVersion
View	LinkViewCreated

Chapter

20

Using Charts

The Windows and Cross-Platform clients offer a wide variety of charts. You can filter out data in the upper pane to display only the data that you want to include in your chart. In addition, you can select specific items from the filtered data to include in your chart.

Charts are created from the data displayed in the upper pane. A maximum of 60 fields can be displayed in the upper pane.

Available Charts

You can display a number of different charts, depending on the selected tab and chart menu option. The different types of charts that can be created include:

- Simple charts
- Distribution charts
- Correlation charts:
- Time-series charts

For audit entries, charts include:

- Distribution charts
- Time-series charts

The following table provides the fields available for creating charts. From the *Show Fields* dialog, you can select the Show Advanced Fields check box to display

additional fields which can be included in your charts. See Chapter 5, “Managing Data” for information on changing column header fields.

Table 20.1 File Charts: Available Fields

Simple	Distribution	Correlation	Time-Series
Content Revision	Create charts based on fields currently being grouped.	Content Revision	Configuration Time
DotNotation ID		DotNotation ID	Created Time
EOL Character		EOL Character	Deleted Time
Object ID		Object ID	File Time Stamp at Check-In
Parent ID		Parent ID	Modified Time
Parent Branch Revision		Parent Branch Revision	Sync Local Time Stamp
Parent Revision		Parent Revision	Working File Time Stamp
Project ID		Project ID	
Revision		Revision	
Revision on Disk		Revision on Disk	
Root Object ID		Root Object ID	
Size		Size	
Sync Branch Version		Sync Branch Version	
Sync Content Version		Sync Content Version	
Sync Local Size		Sync Local Size	
Sync on Path to Root		Sync on Path to Root	
Vault Branch Version		Vault Branch Version	
Version		Version	
Working File Size		Working File Size	

Table 20.2 Change Request Charts: Available Fields

Simple	Distribution	Correlation	Time-Series
Attachment Count	Create charts based on fields currently being grouped.	Attachment Count	Closed On
CR Number		CR Number	Configuration Time
DotNotation ID		DotNotation ID	Created Time
Object ID		Object ID	Deleted Time
Parent ID		Parent ID	Entered On
Parent Branch Revision		Parent Branch Revision	Modified Time
Parent Revision		Parent Revision	Resolved On
Root Object ID		Root Object ID	Total Open*
Version		Version	Verified On

*Total Open is available only if the Entered On and Closed On fields are displayed. Its value is the total number of change requests entered that day/week/month minus the total number of change requests closed on that day/week/month. To get a count of the change requests open at any given time, it is best to select the Cumulative Totals check box and not limit the time period.

Table 20.3 Requirements Charts: Available Fields

Simple	Distribution	Correlation	Time-Series
Ambiguities Found	Create charts based on fields currently being grouped.	Ambiguities Found	Created Time
Attachment Count		Attachment Count	Configuration Time
Children Count		Children Count	Deleted Time
CommentID		CommentID	Modified Time
Expected Effort		Expected Effort	End Modified Time
High Effort		High Effort	
Low Effort		Low Effort	
Number		Number	
Object ID		Object ID	
Parent Requirement ID		Parent Requirement ID	
ResponsibleCount		Responsible Count	
Revision Flags		Revision Flags	
Version		Version	

Table 20.4 Tasks Charts: Available Fields

Simple	Distribution	Correlation	Time-Series
Actual Hours	Create charts based on fields currently being grouped.	Actual Hours	Actual Finish
Attachment Count		Attachment Count	Actual Start
Children Count		Children Count	Configuration Time
Estimated Hours		Estimated Hours	Constraint Date
MS Task Unique ID		MS Task Unique ID	Created Time
Object ID		Object ID	Deleted Time
Parent Task ID		Parent Task ID	Estimated Finish
Percent Complete		Percent Complete	Estimated Start
Task Duration		Task Duration	Last MS Project Update
Task Number		Task Number	
Version		Version	Last Work/ Dependency Update
			Modified Time

Table 20.5 Topics Charts: Available Fields

Simple	Distribution	Correlation	Time-Series
Attachment Count	Create charts based on fields currently being grouped.	Attachment Count	Created Time
Children Count		Children Count	Configuration Time
Recipient Count		Recipient Count	Deleted Time
Object ID		Object ID	Modified Time
Parent Topic ID		Parent Topic ID	
Topic Number		Topic Number	
Version		Version	

Table 20.6 Audit Entries Charts: Available Fields

Simple	Distribution	Correlation	Time-Series
Not available	Create charts based on fields currently being grouped.	Not available	Created Time Deleted Time Modified Time

Creating a Simple Chart

A simple chart consists of a series of bars that compare values in one or more fields of selected items. Each field that is compared has its own unique colored bar. To create a simple chart:

1. Select Charts > Simple from the File, Change Request, Topic, or Task menu. The *Simple Chart* dialog appears.
2. Enter the name for the chart in the Chart Name text box.
3. Select one or more fields from the Series list box. The application lists the fields in the upper pane that can be charted. A maximum of 60 fields can be displayed in the upper pane.
4. Do one of the following:
   - Select Current Selection for a chart that displays only the selected items.
   - Choose Select All to display a chart that includes all items from the list.
5. Select a printer page orientation: Portrait or Landscape.
6. Click OK to display the chart.

Creating a Distribution Chart

Distribution charts are based on items in the upper pane that have been grouped together.

Note Fields that are not displayed in the upper pane can also be included with items in the upper pane that have been grouped together.

A distribution chart is represented in the form of a pie chart, where each wedge indicates what fraction of the whole a group represents. For example, you might want to group all change requests based on status.

For information about grouping items based on their values in specific fields, see "Sorting and Grouping the Data" on page 54. You can group based on the values in a maximum of four fields.

To create a distribution chart:

1. Select Charts > Distribution from the File, Change Request, Topic, Task, or Audit menu. The *Distribution Chart* dialog appears.
2. Enter a name for the chart in the Chart Name text box.
3. Do one of the following:
   - Select Current Selection for a chart that displays only the selected items.
   - Choose All to display a chart which includes all items from the list.
4. Select a printer page orientation: Portrait or Landscape.
5. Click OK to display the distribution chart.

Creating a Correlation Chart

A correlation chart is represented in the form of a scatter chart. You select the fields that represent the x-axis and y-axis. The application plots the points on the chart showing the correlation between the two axes.

To create a correlation chart:

1. Select Charts > Correlation from the File, Change Request, Topic, or Task menu. The *Correlation Chart* dialog appears.
2. Enter the name for the chart in the Chart Name text box.
3. Select one axis label from the x-coordinates list box. The coordinate list boxes display the names of the fields that are displayed in the upper pane that can be used as axes. A maximum of 60 fields can be displayed in the upper pane.
4. Select another axis label from the y-coordinates list box.
5. Do one of the following:
   - Select Current Selection for a chart that displays only the selected items.
   - Choose All to display a chart which includes all items from the list.
6. Select a printer page orientation: Portrait or Landscape.
7. Click OK to display the correlation chart.

Creating a Time-Series Chart

A time-series chart is a line chart showing the number of items that have the same day, week, or month in the specified time/date field.

To create a time-series chart:

1. Select Charts > Time Series from the File, Change Request, Topic, Task, or Audit menu. The *Time-Series Chart* dialog appears.

Note A time-series chart must be based on a time/date field. If the upper pane displays no time/date fields, the application displays an error message.

2. Enter a name for the chart in the Chart Name text box.
3. (Optional) Select the Cumulative Totals check box to display the accumulated items in the chart.
4. (Optional) Select the Include Non-Represented Dates check box to include all date ranges in the chart.
5. Select Daily, Weekly, or Monthly from the Tracking Interval box.
6. Select one or more items from the Time Series list box.
7. (Optional) Select the Limit To Period Of Time check box to include a specific date range for your chart.
8. Do one of the following:
   - Select Current Selection for a chart that displays only the selected items.
   - Choose All to display a chart which includes all items from the list.
9. Select a printer page orientation: Portrait or Landscape.
10. Click OK to display the time series chart.

Using the Chart Window in the Windows Client

The chart window, by default, displays a toolbar and a palette bar. It may, optionally, display a pattern bar. Here's a list of the charting toolbar buttons and their functions:

Import Chart
Open a chart file that has been previously saved to disk.

Export Chart
Save a chart file to disk.

Copy to Clipboard as Bitmap
Copy the chart to the Windows Clipboard as a bitmap.

Copy to Clipboard as Text
Copy the chart data to the Windows Clipboard as text.

Print Chart
Print the chart.

Change Gallery Type
Open a drop-down list box that enables you to change the type of chart (bar, scatter, pie, etc.). The button changes to indicate the current selection.

Change Color
Open a drop-down palette that enables you to change the active color. The color can then be dragged to any chart element that accepts color.

Switch between 3D and 2D
Toggle between three-dimensional and two-dimensional charts.

Rotate Chart
Open a dialog that enables you to change the viewing angle and perspective of a three-dimensional chart.

Z-Clustered Series
Add depth to a multi-series chart.

Zoom
Enlarge a specific area of the chart.

Show or Hide Legend
Toggle the display of the chart legend on and off.

Show or Hide Series Legend
Toggle the display of the series legend on and off.

Vertical Grid
Toggle the display of a vertical grid on and off.

Horizontal Grid
Toggle the display of a horizontal grid on and off.

Edit Titles
Edit title text.

Change Text Fonts
Select a font for chart labels and legends.

Tools
Open a drop-down list box that enables you to toggle display elements such as the toolbar, palette bar, and pattern bar on and off.

Change Chart Options
Open a multi-tab dialog in which you can set chart type and appearance, scale, view, and title text.

Editing Charts

After you generate a chart, you can change its appearance, change the chart data, copy the chart as a bitmap or as text, and print the chart.

Changing a Chart Type

The application generates a chart type appropriate for the selected data. Available chart types include pie, bar, scatter, area, surface, and more. You can change the chart type to suit your needs.

To change the chart type:

1. From the toolbar, click the Change Gallery Type drop-down list box. A drop-down value set of possible types opens.
2. Select a chart type. Each chart type has a tool tip explaining the type of chart represented by the picture.

Changing Chart Data

After you generate a chart, you can change the data to perform “what if” calculations or to fine-tune the chart.

To change the chart data:

1. From the toolbar, click the Tools button. The Tools drop-down menu opens.
2. Select Data Editor from the menu bar. The Data Editor window appears in place of the chart. It lists each value used to create the chart.
3. Edit the values as necessary.
   a. Double-click a value to be changed. The value is highlighted and the alignment changes from right-aligned to left-aligned.
   b. Enter the new value.
   c. Do one of the following:
      - Press tab to move through the data and make changes.
      - Double-click another value to be changed.
   d. Repeat step c as often as necessary until you are finished editing the data.
4. From the toolbar, click the Tools button. The Tools drop-down menu opens. There is a check mark next to the Data Editor item.
5. Select Data Editor from the menu bar, to remove the check mark from the menu bar, close the Data Editor window, and regenerate the chart.

Changing Chart Color or Pattern

You can change the color or pattern of any chart element, including:

- Individual slices of a pie chart
- Bars on a bar chart
- Chart backgrounds
- Legend backgrounds

To change the color or pattern of a chart element:

- Do one of the following:
  - Drag the color from the Change Color toolbar button or the color palette to an element that accepts color.
  - Drag the pattern from the pattern bar to a element that accepts pattern.

To change the color on the Change Color toolbar button:

1. From the toolbar, click the Change Color drop-down list box.

   A drop-down palette opens.

2. Select a color from the palette.

The palette closes and the color on the button changes to reflect your selection.

To change the color of points on a line, click the Change Chart Options button on the toolbar, then select the Series tab, then the correct series, and then a color from the Series Color box.

To display the palette bar and/or the pattern bar:

1. From the toolbar, click the Tools button.

   A drop-down menu opens. Items on this menu can be toggled. Selecting an item without a check mark activates the option and places a check mark next to the menu item. Selecting the item again deactivates the option and removes the check mark from the menu bar item.

2. Select Palette Bar or Pattern Bar to display the element.

   While the palette bar and/or pattern bar are displayed, you can use the drag-and-drop technique to copy a color or pattern onto any chart element that accepts color or patterns (more chart elements accept color than patterns).

To remove patterns from chart elements:

1. From the toolbar, click the Change Chart Options button.

   The *Chart Properties* tab dialog appears with the General tab selected.

2. From the toolbar, click the Color Scheme drop-down list box.

   A drop-down value set opens. Color scheme options are:

   - Solid color
   - Black and White pattern
   - Color pattern

3. Select Solid Color to remove any previously applied patterns, then click Apply.

Changing Chart Text

The application automatically provides a top title and a legend for your chart. You may also provide left, right, and bottom title text, edit title or legend text, and change text fonts for each text element on the chart.

To change the legend text:

1. From the toolbar, click the Tools button.

   The Tools drop-down menu opens.

2. Select Data Editor from the menu bar.

   The *Data Editor* window appears in place of the chart. The Data Editor lists each value used to create the chart.

3. Edit the legend text as necessary.

a Double-click a text element to be changed. The text is highlighted.

b Enter the new text.

c Repeat steps a and b as often as necessary until you are finished editing data.

4 From the toolbar, click the Tools button.

The Tools drop-down menu opens. There is a check mark next to the Data Editor item.

5 Select Data Editor from the menu bar, to remove the check mark from the menu bar, close the *Data Editor* window, and regenerate the chart.

To add or change title text:

1 From the toolbar, click the Edit Titles button.

The *Titles* dialog appears.

2 Enter title text (or edit existing text) for top, bottom, left, and/or right titles.

3 Click Apply to view your changes before closing the dialog, or click OK.

To change font or color for any text element:

1 From the toolbar, click the Change Text Fonts button.

A drop-down menu listing all available text elements opens.

2 Select the text element for which to change fonts.

A standard *Font* dialog appears.

3 Make changes to the font, style, point size, emphasis, and/or color, as appropriate.

Changing Viewing Options for 3D Charts

You can view charts in two dimensions (2D), three dimensions (3D), or full 3D view. In full 3D view, you can tilt and swivel the chart.

To switch between 2D and 3D:

- From the toolbar, click the Switch Between 2D and 3D Views button.

To tilt and swivel a 3D chart:

1 From the toolbar, click the Rotate Chart button.

The *3D View Properties* dialog appears.

2 Select the Full 3D View check box, if necessary, to activate full three-dimensional viewing options.

3 Using the mouse, drag the blue dot around the x-axis plane and/or drag the red dot around the y-axis plane.

Exploding a Pie Chart

To emphasize one or more values, a pie chart is frequently “exploded”. That is, one or more slices of the pie are shown separated from the rest.

To explode a pie chart:

1 Click the slice to be separated from the rest of the pie. The mouse pointer changes to a cross.

2 Drag the slice away from the center of the pie chart.

Using the Chart Window in the Cross-Platform Client

The Chart window displays only after you create a chart. To do this, click a component tab, display data in the upper pane, select Charts from the component menu, specify the type of chart you wish to view, and complete the Chart dialog.

The Chart window in the Cross-Platform client consists of the following:

- Title Bar: The title bar at the top of the main window shows the following information:
  - Product name
  - Name of the chart
- Toolbar: The Chart window has its own, unique toolbar. For more information on the toolbar and its features, see “Using the Chart Window Toolbar” on page 268.
- Chart Contents: The type of chart and its display vary considerably, according to your choices. The chart can include not only data but one or more titles, legends, and vertical or horizontal grids.

Using the Chart Window Toolbar

The chart window toolbar for the Cross-Platform client includes the following options. Most are self-explanatory buttons that work as toggles. Other buttons present a dialog for you to make choices or enter your own values. From left to right, the options are as follows:

- Save chart as: For more information, see “Exporting Chart Images” on page 268.
- Change chart type: For more information, see “Changing Chart Type” on page 269.
- Edit colors: For more information, see “Editing Chart Colors” on page 270.
- Toggle 3D view: For more information, see “Viewing a Chart in 3D or 2D” on page 271.
- Toggle legend: For more information, see “Showing Legends on Charts” on page 272.
- Toggle Vertical grid: For more information, see “Showing Horizontal or Vertical Grids on Charts” on page 272.
- Toggle Horizontal grid: For more information, see “Showing Horizontal or Vertical Grids on Charts” on page 272.
- Edit titles: For more information, see “Editing Titles on Charts” on page 272.
- Toggle table: For more information, see “Viewing Chart Data” on page 272.

Exporting Chart Images

Using the Cross-Platform client, you can export a chart to a JPG file.

To export/save a chart as a JPG file:

1. Click the **Save chart as** button on the toolbar. The Save As dialog opens.
2. Enter a name for the file.
3. Browse to or enter a path where you wish to save the JPG file.
4. Click Save.

Changing Chart Type

To change to a different chart type, click the combo box drop-down arrow, and choose a chart type from the drop-down list. The drop-down list displays the icon and name for the all of the available chart types.

The table below lists the available chart types in the Cross-Platform client and their descriptions.

Table 20.7 Cross-Platform client chart types and descriptions

Chart type	Description
Area	Area charts emphasize the amount of change over a period of time or compare multiple items. An area chart also shows the relationship of parts to a whole by displaying the total of the plotted values. An area chart is a form of line chart, but the area between the horizontal (X) axis and the line connecting the data makers is filled with color. This makes it easy to see where the points encompassed by the different data series overlap.
Bar	Bar Charts show the changes in a data series over time or compare multiple items. Types of items are arranged vertically and data values are plotted horizontally to emphasize variation over time. The 3-D bar chart provides an extra dimension for plotting data by comparing values along two axes.
Bubble	Bubble charts are a type of scatter chart. The x and y coordinates of the data marker (the bubble) are determined by two data values. The size of the data marker indicates the value of a third variable.
Column	These charts show the changes in a data series over time or compare multiple items. Types of items are arranged horizontally and data values are plotted vertically to emphasize variation over time. The 3-D bar chart provides an extra dimension for plotting data by comparing values along two axes.
Heat Map	Shows the relationship between data items by using gradually changing shades of color. Heat map charts are commonly used in financial analysis to show which stocks are rising, which are falling and the amount and rate of change between them.
Line	Emphasizes the amount of change over a period of time or compares multiple items. Data points are plotted in series using evenly-spaced intervals and connected with a line to emphasize the relationships between the points.
Pie	Shows the size of items that make up a data series proportional to the total of the items in the series. A pie chart always shows a single data series and is useful for determining which items in the series are most significant.
Scatter	Used either to show the relationship among the items in several distinct series of data, or to plot two sets of values as one series of x/y coordinates. A scatter chart draws attention to uneven intervals or clusters of data. This type of chart is often used to plot scientific data, and can highlight the deviation of collected data from predicted results.
Stack Bar	Shows the relationship of individual items in a series to the whole.
Stack Column	Shows the relationship of individual items in a series to the whole.

The table below lists the available chart types in the Cross-Platform client for each respective chart (simple, distribution, correlation, time-series) that you can choose in the Change chart option of the Chart window.

Table 20.8 Cross-Platform client chart types per chart

Chart	Chart type
Simple	Bubble
	Column
	Heat map
	Line
	Pie
	Scatter
	Stack bar
	Stack column
Distribution	Area
	Bar
	Column
	Heat map
	Line
	Pie
	Stack bar
	Stack column
Correlation	Area
	Bar
	Column
	Heat map
	Line
	Pie
	Scatter
	Stack bar
	Stack Column
Time Series	Area
	Bar
	Column
	Heat map
	Line
	Bar
	Stack bar
	Stack Column

Editing Chart Colors

Using the Cross-Platform client, you can Edit the colors of your series data.

To change the colors of series data from the Chart window toolbar:

1. Click **Edit Colors** on the Chart window toolbar. The Edit Colors dialog opens.
2. Check *Use Custom Colors*.
3. Select a series from the list, and click **Edit**. You can edit one series at a time. The Select Color dialog opens.
4. Using the tabs at the top of the Select Color dialog, you can choose your own custom color for the selected series. You can choose from swatches of color or from

HSB or RGB values. Make you desired selection, and click **OK** to exit the Select Color dialog.

5 In the Edit Colors dialog either:

- Select another series to edit its color, or
- Click **OK** to apply your changes and exit the Edit Colors dialog.

Viewing a Chart in 3D or 2D

The Cross-Platform client displays the majority of charts by default in 3D (see table below for more information). You also have the option to toggle the 3D/2D view in the Cross-Platform client.

To toggle between a 3D and a 2D chart:

1 Display a chart.

2 Click the Toggle 3D View button on the Chart window toolbar to create a 2D view of your data.

3 To return to a 3D chart, click the Toggle 3D View button again.

Zooming and Rotating Charts

You can zoom in and out and rotate a chart on its x-y-z axes for all charts that you can view in 3D view. The majority of the charts shown in 2D view do not allow for zoom or rotate operations. Table 20.9 summarizes these points and gives information about the default view for each chart type in the Cross-Platform client.

To zoom in and out on a chart:

1 Display a chart. Be sure that the chart type selected is in a view that allows for zooming. Refer to the chart below to see which charts allow zooming.

2 While holding down the CTRL key, move your mouse from left to right to zoom in and out on the chart.

To rotate a chart:

1 Display a chart. Be sure that the chart type selected is in a view that allows for rotating. Refer to the chart type below to see which charts allow rotating operations.

2 While holding down the SHIFT key, move your mouse in a circular motion to rotate the chart.

Table 20.9 Default views and zoom/rotate capabilities for Cross-Platform client chart types

Chart	3D - Zoom, rotate	2D - Zoom, rotate	Default view
Area	Yes	No	3D
Bar	Yes	No	3D
Column	Yes	No	3D
Bubble	No	No	2D (3D view not available)
Heat Map	Yes	No	3D
Line	Yes	No	3D
Pie	Yes	No	3D
Scatter	No	No	2D (3D view not available)
Stack Bar	Yes	No	3D
Stack Column	Yes	No	3D

Showing Legends on Charts

Using the Cross-Platform client, you can display or hide legends.

To toggle a legend display for your chart:

1. Display a chart.
2. Click **Toggle legend** on the Chart window toolbar to remove the legend from view.
3. To show the legend, click **Toggle legend** again.

Showing Horizontal or Vertical Grids on Charts

Using the Cross-Platform client, you can optionally display horizontal or vertical grids on charts.

To toggle a horizontal grid for your chart:

1. Display a chart.
2. Click **Horizontal grid** on the Chart window toolbar to remove horizontal grid lines from the chart.
3. To add horizontal grid lines to the chart, click **Horizontal grid** again.

To toggle a vertical grid for your chart:

1. Display a chart.
2. Click **Vertical grid** on the Chart window toolbar to remove vertical grid lines from the chart.
3. To add vertical grid lines to the chart, click **Vertical grid** again.

Editing Titles on Charts

When you create a chart, you can insert titles for the left, right, top, or bottom of the chart. These titles can serve as overall titles or names for the axes.

To create or edit titles for your chart in the Cross-Platform client:

1. Click **Edit Titles** on the Chart window toolbar. The Edit Titles dialog opens.
2. Enter or edit the contents of the Top, Left, Right, or Bottom text boxes.
3. Click **Apply** to make changes to the chart and keep the Edit Titles dialog open.
4. Click **OK** to close the dialog.

Viewing Chart Data

After you display a chart, you can use the Toggle Table button in the Chart window toolbar to display the data on which it is based in a table format.

To display chart data:

1. Click **Toggle Table** on the Chart window toolbar.The chart data displays in a matrix that shows the legend entries vertically and the charted values for each of the entries horizontally.
2. Click **Toggle Table** again. The chart displays in the Chart window.

Chapter

21

Setting Personal Options

The application enables you and your team members to set personal options that suit your individual work styles. These options apply to the currently logged-on user on a given workstation. You can also update the list of servers available to your workstation and update your user account on the server on which you are currently logged in.

Customizing the Application

You can decide what confirmation messages will be shown, whether shortcuts and folder selections will be remembered, control the toolbar and status bar, and so on.

To customize the application:

1. Select Tools > Personal Options from the menu bar. The *Personal Options* dialog appears.
2. Select the Workspace tab.
3. Use the Confirm check boxes to confirm deletions, moves, shares, and warnings.
4. Use the Display check boxes to display or hide the toolbars and the status bar.
5. Select the Show History Times As UTC check box if you would prefer to see the times in the Time column of the History pane as UTC times rather than times in your local time zone. UTC times end in “Z” to differentiate them from local times.

   The Z stands for the “zero meridian”, which goes through Greenwich in London. It is also commonly used in radio communication where it is pronounced “Zulu” (the word for Z in the international radio alphabet).
6. If you select the Restore Folder Selection On Tab Change check box, the application remembers what folder was selected the last time you used a particular tab and returns to that folder when you reselect the tab. If you have not selected a particular tab this session, the application automatically selects the view’s root folder when you switch to that tab.

   Otherwise, the selected folder does not change as you switch from tab to tab.
7. Use the Reset Scope to Local on Folder Change check box to automatically reset All Descendants every time you change a folder. This saves you the time the application would take to scan items. Clear this check box so that only you change

All Descendants. (All Descendants is both a button on the toolbar and a command on the File, Change Request, Task, Topic, and Audit menus.)

8 Select the Maintain Group State On Folder/Scope Change check box to keep your place in the upper pane even when you change folders or reset All Descendants. For example, if a particular group is open, it will remain open as you select a different folder. Clear this check box to collapse all the groups as you make a folder or scope change.

9 Use the Automatic Refresh with a Maximum Delay of __ Minutes text box to set the maximum number of minutes between refreshes of your project.

This refresh is equivalent to pressing *Shift+F5*. It updates the folder hierarchy and the upper pane.

10 If you have opened project view windows, you can select the Restore Workspace at Startup check box to reopen any view or view comparison windows that were left open as you exited the application the previous time.

11 Enter or browse for a name and path for your report files in the Report Output Path text box. This path becomes the default location for all reports that you create using the application.

12 Use the Look and feel drop-down list box to select a different skin for the Cross-Platform client. This updates the look and feel of the user interface.

13 The Font adjustment drop-down list box provides the option to increase or decrease the size of the text displayed within the user interface of the Cross-Platform client.

14 Use the application log:

a Enter or browse for a name and path for your StarTeam.Log files in the Log Output Path text box. The default is the folder where the application is installed. The current log file is always named StarTeam.Log. Log files from earlier sessions on your workstation have names that include the date and time the file was last modified.

```
StarTeam-12-Mar-01-09-45-04.Log
```

The StarTeam.Log file contains data about operations sent from your workstation to one or more servers, depending on what project views you have open. The data includes the name of the project so that you can isolate data for a particular server when necessary. To review StarTeam.Log, select Tools > View StarTeam Log. For additional information, see your administrator or the *StarTeam Administator's Guide*.

b Select the Log Errors check box to record errors that occur while you are using the application.

For example, you might see:

```
...Operation 40956 failed: TCP/IP Socket Error 10054:...
```

If you are logging both errors and operations, the application also logs the operation that the server was performing at the time of the error.

c Select the Log Operations That Take At Least __ Milliseconds check box to record operations and/or events.

1 (Optional) Select a number of milliseconds.

The milliseconds time setting stops the log from filling up with operations and events of little importance. The default, 10 milliseconds, is a reasonable setting.

2 Do one of the following:

- Select the Summary option button to include a breakdown of the time spent on the client and the Server for each operation.
- Select the Details option button to list commands along with the summary. Only with the Details option button is the server identified by address or name.

d Select the Log StarTeamMPX Events check box.

For example, you might see:

```
...Statistics for Events /STEvent/01b21d208-51ea...
```

15 Do one of the following:

- Click OK to save your options and close the *Personal Options* dialog.
- Select another tab of the *Personal Options* dialog.

Setting File Options

You can set a variety of file options for locking files, checking them out, controlling where status information is stored, using applications of your choice to edit, compare, and merge them, and so on.

To set file options:

1 Select Tools > Personal Options from the menu bar. The *Personal Options* dialog appears.

2 Select the File tab.

3 Select any of the following *check-out options* for files:

a Select the Use Last Modification Time For Checked-Out Files check box, so that the time stamp for each checked out file is the time stamp of the revision that is being checked out. Otherwise, the checked-out file's time stamp is the current time—the time of the check-out operation.

When you change the setting for this check box, the statuses of files that are already in your working folder can affect the outcome of subsequent check-out operations on those files. For example, if you check out a file while this check box has one setting, and then change the check box's setting, only a force check-out will change the time stamp on that file. This is because the file is considered Current and, therefore, not checked out at all unless you do a force check-out.

The application does not rely exclusively on your computer's clock and the date/time stamp of your working files to determine what needs to be checked in or out. It will still function properly, even if your computer's clock is not in sync with other team members' clocks. However, most compilers and development environments are still sensitive to the date and time stamps. Files that were checked in by another member of your team would, by default, have the date and time stamp set by your teammate's clock, even when you check the file out. If your clock is not in sync with your teammates, you may experience problems compiling or building the file. Therefore, we still recommend using whatever service your network server supports for synchronizing all workstation clocks.

Windows network users can execute the following to synchronize their workstation time with their server:

```
Net time \\server_Name/set/yes
```

b Select the Optimize for Slow Connections check box, so that the Server will send the difference between the currently checked-out revision (in the working folder) and the revision being checked out when that will speed up the data transfer.

When cleared, the server always sends the requested revision in its entirety.

4 Select any of the following *locking options* for files:

a Select Exclusively Lock Files On Check-out so that the Check Out dialog's lock status option will default to Exclusive. (Otherwise, the default is Keep Current.)

b Select Clear File Locks On Check-in so that the *Check In* dialog's lock status will default to Unlock. (Otherwise, the default is Keep Current.)

c Select the Use Non-exclusive Locks In Integrations check box to affect how files are locked when accessed from the application integrations. When you select this check box, locking a file (for example, as part of a Get or Check Out operation) creates a non-exclusive lock rather than an exclusive lock.

With an exclusive lock, only the person who has the file locked can check in the file. With a non-exclusive lock, others can check in the file. Exclusive locks are the safest, but non-exclusive locks are often preferred because text files can be easily merged using Visual Merge. Using non-exclusive locks allows more than one person to edit a file at one time. If team members are not editing the same lines of the file, the merged file usually has no conflicts.

If you are using an application integration for your development environment (for example, the integration with Visual Studio .NET), you cannot check in files from the development environment if both the Require Exclusive Lock When Files Are Checked In check box on the *Project Properties* dialog (Options tab) and the Use Non-exclusive Locks In Integrations check box on the *Personal Options* dialog (Files tab) are selected. The administrator usually determines the setting of the Require Exclusive Lock When Files Are Checked In check box. However, personal options are set by you for your workstation.

If you have set the Use Non-exclusive Locks In Integrations check box and experience check-in problems, try clearing the check box. You may want to talk to your administrator about the setting for the Require Exclusive Lock When Files Are Checked In check box.

d Select the Mark Unlocked Working Files Read-Only check box, to have working copies of the files that you change from locked to unlocked become read-only when you perform the following file operations:

- File check-in
- File check-out (from file or history list)
- File unlock

This way, only locked files can be edited.

This option is identical to a project property. When selected, the project property overrides the personal option. It applies to files that are unlocked either from the application or from the integrations with third-party applications.

Caution If you select this personal option (or project property) and then change your mind, make sure that no files are read-only *before* you change the setting from selected back to cleared. Before changing to cleared, you can forcibly check out all the files (or all the read-only files—if you know which ones they are) as locked files. Then unlock them. When this option is cleared, you must use the operating system to change the read-only attribute to read/write.

5 Select any of the following *merging options* for files:

a Always Pop Up Merge Utility, to view the merged file—even when no conflicts are found.

b Pop Up Merge Utility In Case Of Conflicts Only, to view the merged file only if it contains conflicts. Otherwise, files are checked in and out automatically.

6 Select any of the following *general options* for files (available only for the Windows client):

a Select the Use File Checksums (MD5) To Calculate Status check box, to use a file's checksum instead of its time stamp (and size) to compute the Status field when the application is refreshed. The checksum provides a more accurate status value but takes longer. For example, with a time stamp-based status, a file's status may be Modified when only its time stamp has changed. (You might have opened a file in an editor and saved it even though you have made no changes.) Using the checksum results in a status of Current because the local file checksum would match the tip revision of the file.

   Sometimes two files have the same time stamp and size, even though the files are different. Again, to get the correct status information in this case, use the MD5 checksum.

b Select the Automatically Detect Changes To Working Files check box, to monitor local file changes. When selected, the application becomes aware of local file status changes earlier. The application does *not* display the changes until either a manual or automatic refresh occurs. However, the refresh is faster because the application already has the local file status information needed to complete the refresh.

   Clear the check box to stop the application from monitoring local file changes.

7 Select any of the *EOL options* for files (available for the Cross-Platform client). The Cross-Platform client assumes that you are working from a specific platform either always or for lengthy periods of time. It also assumes that you want to ensure that any file on your workstation has the EOL character that you need to work on that platform. As you add files or check them in or out, the EOL characters within the files can be converted to the EOL character of your choice. The personal options for EOL characters enable you to select default settings for those operations.

   In the Windows client, EOL options are set per folder for all users of that folder. That client assumes that the folder you are using is mapped to a different platform than your workstation's platform. See "Creating Folders for Other Operating System Files" on page 46 for more information.

   As you add files or check them in, EOL characters are always converted to CR/LF. When you compare files, EOL characters are ignored. When you update status, the EOL character is always converted to CR/LF as well.

   To set the EOL character for adding and checking in files:

   - Select or clear the "Automatic EOL conversion (CR-LF) for add/check-in operations" check box.

   To set the EOL character for checking out files:

a Select or clear the "Automatic EOL conversion for check-out operations" check box.

b Select the appropriate option button for the EOL character.

Note Depending upon the application, Mac System X files may have LF or CR as the EOL character.

8 Select a default file encoding for keyword expansion. The selection made here will be the default file encoding for file check out.

9 Select *File status repository options* for files:

   File status information is stored on your workstation. By default, the information is stored at a central location. The default location is a folder named syncdb, but its path differs depending on the your workstation's platform.

   - For Windows NT, syncdb is a child of: *C:\Winnt\Profiles\username\Local Settings\Application Data\Borland\StarTeam*.

- For Windows 2000 or XP, syncdb is a child of:
  C:\Documents and Settings\username\Local Settings\ Application Data\Borland\ StarTeam
- For non-Windows platforms, syncdb is a child of:
  /user_home_directory/.starteam-client/

Each syncdb folder contains folders that represent your working folders (although their names appear to be unrelated) and contain a .sdb and .lck file.

You can change the central location or, if you prefer, specify per-folder storage. Then each working folder has file status information stored in a child folder named .sbas. Each .sbas folder contains a Sync and a SyncLock file.

a. To control where status information is stored:

   1. Specify the type of storage by selecting the Central or Per Folder option button.
   2. (Optional) If you selected Central, you can enter or browse for a location other than the default. The new location goes in the Central Repository Location text box. In this case, the term repository does not refer to your database but to a location for storing file statuses.

b. to delete old and irrelevant status information:

   1. Click Purge to clean up the file status repository. The *Status Repository Cleanup* dialog appears.
   2. Click Start to see a list of paths to folders that no longer exist.
   3. Click Purge to delete the status information for selected folders, or click Purge All to delete all the superfluous status information.

You can use a different storage method for a specific view by setting view properties. See “Controlling File Status Information” on page 24 for more information.

10. In *URL Options* : Both URL options can be set for change requests, requirements, tasks and topics, as well as for files. Different settings can be selected for different components.

- In the Display Template text box (applies only to files and other items, not to folders), type the template to be used to generate an HTML representation of an item when the item’s URL is copied to the Clipboard. With no format, there a default HTML representation that specifies the type of item and identifies it by name and number. For example, “Change Request #34, 132.”

  When the text is generated from the template, the specified property values are substituted for the variables in ~~*~~. The variables may be referenced by the same names used in report templates, as well as by the display name of the property. When using the display name, you can omit spaces, and case will be ignored.

  For example, if you use the following sample template for a file:

  ```
  ~~FolderPath~~~Name~~
  ```

  the HTML representation will be the StarTeam path to the selected file. This template is a superset of that used by the Report feature of the client.

- Select or clear Generate ID-based URLs. When this option is selected, the URL will be specified by ID rather than by name. For example, an ID-based URL would be starteam://hostname:49201/12;ns=Project;scheme=id, while a name-based URL would be starteam://hostname:49201/myproject.

Note As folders do not have a Personal Options tab, there is no way to set the URL type for a folder. Therefore, folders always use ID-based URLs.

11 On operating systems other than Windows, you can set Alternate Applications to display files in a specified application.

If the client is installed on a Windows system, double-clicking a file in the upper pane displays that file in an associated application. Because there is no standard way to display a file on non-Windows platforms, the Cross-Platform client offers a personal option to resolve this problem. (You never see this option if the client is running on a Windows system.)

The Open With text box enables you to provide a command that will display at least one type of files. The command should consist of the path to an application and the command-line options for which the application can substitute the selected file.

The application runs this command whenever you do one of the following:

- Double-click a file in the item list
- Double-click an attachment
- Generate and open a report

The following command is suggested:

```
netscape -remote "openFile($file)"
```

because Netscape can handle many different media types, such as images files, text files, and html.

To associate a specific type of file:

a Click Browse to locate the command.

b Add the appropriate options.

The application will substitute the selected file for the $file in the command.

12 To set *alternate application options* for a text editor, comparison utility, and/or merge utility:

a Click Alternate Applications. The *Alternate Applications* dialog appears.

b Select the Editor check box, to use an editor other than Notepad.

1 Enter the path to or browse for that editor in the text box.

2 (Optional) In the Options text box, add the command-line options to be used with that utility. For example, if you are using the CodeWright editor, you might use -nosplash to hide the splash screen.

c Select the Merge Utility check box, to use a merge utility other than Visual Merge.

1 Enter the path to or browse for that merge utility in the text box.

2 In the Options text box, add the command-line options that go with that utility. Use the following terms:

$branchtip: A place holder for the path to the tip revision of the file to be merged.

$usertip: A place holder for the path to the local working file to be merged.

$basefile: A place holder for the path to the common ancestor for the $branchtip and $usertip files.

$resultfile: A place holder for the path to the file that will store the output from the merged file.

For example, suppose the Merge Utility text box contains “D:\Programs\Merge Utility 5\Mergeutil.exe” and the Options text box contains “-s $branchtip $usertip $basefile $resultfile”. The application assumes that your options are appropriate for the utility to be used. Each time that the application needs to call the merge utility, it replaces the terms that start with $ with actual paths to those files and asks the operating system to execute the completed command.

d Select the Comparison Utility check box, to use a comparison utility other than Visual Diff.

1 Enter the path to or browse for that comparison utility in the text box.

2 In the Options text box, add the command-line options that go with that utility. Use the following terms:

$file1: A place holder for the path to the first of the two files to be compared.

$file2: A place holder for the path to the second of the two files to be compared.

For example, suppose the Comparison Utility text box contains "*D:\Programs\Comparison Utility 8\Diffutil.exe*" and the Options text box contains "`$file1 $file2`". The application assumes that your options are appropriate for the utility to be used. Each time that the application needs to call the comparison utility, it replaces the terms that start with $ with actual paths to those files and asks the operating system to execute the completed command.

13 Click OK to save your options and close the *Personal Options* dialog.

Managing Read/Unread Items

Items that you have not read are listed in bold in the upper pane if you are:

- The user responsible for the change request, requirement, or task.
- A recipient of the topic.

This feature does not apply to files and audit entries.

After you display their properties, the bold disappears.

You control additional conditions under which the bold disappears. An item can change from read to unread:

- If it is selected for some period of time
- When you move from the item by selecting another item
- Only when you manually select the menu command that marks the item as read or unread

To manage read/unread items:

1 Select Tools > Personal Options from the menu bar. The *Personal Options* dialog appears.

2 Select the Change Request, Requirement, Topic, or Task tab.

3 Select any of the following Mark as Read options:

- Select the When Change Request (or Task or Topic) Is Selected option button, to indicate that an item is read as soon as you select it.
- Select the When Selected For __ Seconds Or More option button and enter a number, to indicate that an item has been read after a designated time interval. The range is from 15 to 9999 seconds.
- Select the Only When Manually Marked As Read option button, to mark the item read only by using the menu option rather than as a result of selection.

The item is always marked as read when you display its properties.

4 Click OK to save your options and close the *Personal Options* dialog.

Controlling System Tray Notification

While you are running the application, you can check for changes items that may affect you. This feature does not apply to files and audit entries. The application notifies you when:

- A change request, requirement, or task becomes your responsibility or a topic names you as a recipient.

  If a topic has no recipients listed, no one receives notification.

- A requirement or task that is your responsibility or a topic for which you are a recipient has changed.

In the Windows client, the defect, requirement, task, or topic icon displays on your workstation's task bar (in the system tray).

In the Cross-Platform client, the icons appear at the right end of the status bar, rather than in the system tray.

Double-clicking the icon opens the *<project>:New <item type>* dialog which lists the items that need your attention. Double-clicking the item displays its properties. After you have opened and, hopefully, read the item's properties, the item disappears from the dialog. You might want to take notes while displaying the item's properties.

Note If your administrator has enabled e-mail notification, you will automatically receive e-mail messages notifying you about change requests for which you are responsible, about changes in any requirements and tasks for which you are responsible, and about changes in any topics for which you are a recipient. E-mail notification is client independent and you do not need to run the application to receive notifications. You can, however, use the system tray notification with or without e-mail notification.

To control system tray notification:

1. Select Tools > Personal Options from the menu bar.

   The *Personal Options* dialog appears.

2. Select the Change Request, Requirement, Topic, or Task tab.

   The contents of this tab looks slightly different in the Windows and Cross-Platform clients.

3. In the Windows client:

   a. To be notified about unread items for which you are responsible or for which you are a recipient, select the Check for New Or Modified <items> Every __ Minutes check box.

   b. Enter the time interval between checks for items that need your attention. The default is 10 minutes.

   c. Go to step 5.

4. In the Cross-Platform client:

   a. To be notified about unread items for which you are responsible or for which you are a recipient, select the Check for New Or Modified <items> check box.

   b. In the Interval text box, enter the time interval between checks for items that need your attention. The default is 10 and the interval is in minutes.

   c. Go to step 5.

5. Click OK to save your options and close the *Personal Options* dialog.

Automatic Locking of Items Other than Files

Two personal options control locks for items other than files. Use them to do the following:

- Lock an item as you open its *<item> Properties* dialog. (The application assumes that you plan to edit it.)
- Unlock an item as you click OK to exit its *<item> Properties* dialog. This option unlocks the item as a new revision of the item is created.

If you select neither option, opening an item does not lock it. You manually control all locking and unlocking of items.

If you select only the first of these two options, items that you have not already locked become locked as you open them and unlocked as you click Cancel or OK.

If you select only the second option, any item that you have locked manually becomes unlocked as you click OK to create a new revision.

If you select both of these options, you can lock items manually or by opening them. They become unlocked when you click OK to create new revisions. If they were not locked prior to being opened, they also become unlocked when you click Cancel.

To automatically lock and unlock items:

1. Select Tools > Personal Options from the menu bar.

   The *Personal Options* dialog appears.

2. Select the Change Request, Requirement, Topic, or Task tab. (This does not apply to files, for which the locking options are slightly different. See "Select any of the following locking options for files:" on page 276.)

3. To lock unlocked items as you open their properties dialogs, select the Exclusively Lock <Item> On Edit check box.

4. To unlock locked items as you create a new revision for them, select the Clear <Item> Locks After Edit check box.

5. Click OK to save your options and close the *Personal Options* dialog.

Using StarTeamMPX

To take advantage of the caching services and performance enhancements offered by the Servers that use StarTeamMPX, StarTeamMPX must be enabled on your workstation. To check on the current status of StarTeamMPX on your workstation, check the right end of the application status bar.

The following table describes the icons and words that provide information about StarTeamMPX when they appear on the status bar.

Table 21.1 Status Bar Information about StarTeamMPX

Status Bar Information	Description
Yellow lightning bolt	Indicates that StarTeamMPX is available and enabled for the currently selected project view.
Grey lightning bolt	Indicates that StarTeamMPX is available for the currently selected project view but that it has not been enabled.
Disabled lightning bolt (diagonal line)	Indicates that StarTeamMPX was enabled for the currently selected project view but something happened to break the connection. Perhaps the network is not working properly.
No icon	Indicates that StarTeamMPX is not available for the currently selected project view.
Instant	Indicates that StarTeamMPX's autorefresh is turned on.

Table 21.1 Status Bar Information about StarTeamMPX (continued)

Status Bar Information	Description
Auto	Indicates that your workstation's autorefresh is turned on—but that StarTeamMPX's autorefresh is either turned off or unavailable. Your workstation's autorefresh is on the Workspace tab of the *Personal Options* dialog.
Manual	Indicates that your workstation's autorefresh is turned off and that StarTeamMPX's autorefresh is either turned off or unavailable.

By default, StarTeamMPX is enabled on your workstation.

To use the StarTeamMPX on your workstation:

1. Select Tools > Personal Options from the menu bar. The *Personal Options* dialog appears.
2. Select the StarTeamMPX tab.
3. Select the Enable StarTeamMPX check box, to use StarTeamMPX.
4. Do one of the following:
   - To refresh manually (*Shift+F5*), clear the Automatic Refresh With check box.
   - To refresh automatically:

     After every change, the application waits a minimum number of seconds before refreshing. That means that if changes are infrequent, the application performs a refresh almost immediately. However, if changes are frequent, the minimum refresh timer is constantly being reset and never reaches the number of seconds set for a refresh. In such cases, the next refresh occurs when the maximum number of seconds between refreshes forces a refresh.

     1. Select the Automatic Refresh With check box.
     2. Set a minimum number of seconds between refreshes in the Minimum Delay Of ___ Seconds text box. The default is 30 seconds.
     3. Set a maximum number of seconds between refreshes in the Maximum Delay Of ___ Seconds text box. The default is 60 seconds.
5. Click OK.

To stop using StarTeamMPX:

- Clear the Enable StarTeamMPX check box to stop using StarTeamMPX.

Changing Personal Information

Occasionally, you will need to update information about your account on the server. For example, you will want to change your password occasionally.

To review your account information:

1. Select Tools > My Account from the menu bar. The *My User Account* dialog appears.
2. Enter or edit any appropriate information.

To change your password:

1. Select Tools > My Account from the menu bar.

   The *My User Account* dialog appears.
2. Click the Logon tab.
3. Do one of the following:

- If your administrator allows blank passwords and you want to use one, select the Set A Blank Password check box.
- In the Password and Confirm text boxes, enter and confirm your new password.

Chapter

22

Using the stcmd Command-line Interface

The application stcmd command-line interface comes in both a COM (Windows) and a Java version. Both versions have the same commands and are installed as part of the StarTeam SDK Runtime. On Windows workstations, the COM version of the command line is installed along with the Windows or Cross-Platform client. For UNIX and other non-Windows workstations, the Java version of the command line is installed along with the Cross-Platform client.

You can perform command-line operations from either of the following:

- A command-line session using the command stcmd and the appropriate options.
- A development environment that enables you to add tools to menus.

A Server must be accessible and running the server configuration for the projects that are to be accessed.

The command-line operations are:

- Add files (**stcmd add**)
- Add folder (**stcmd add-folder**)
- Add project (**stcmd add-project**)
- Add view (**stcmd add-view**)
- Apply labels to selected files (**stcmd apply-label**)
- Check in files (**stcmd ci**)
- Check out files (**stcmd co**)
- Delete files from the working folder (**stcmd delete-local**)
- Display the differences between two revisions of a file (**stcmd diff**)
- Change description (**stcmd dsc**)
- Display revision history of files (**stcmd hist**)
- Create a new label for a view (**stcmd label**)
- Lock or unlock files (**stcmd lck**)

- List files (**stcmd list**)
- Create a working folder (**stcmd local-mkdir**)
- Move file status information from the server to the client as part of upgrading to StarTeam Server 5.2 or later (**stcmd move-status**)
- Remove files from version control (**stcmd remove**)
- Lock and unlock a server (**stcmd server-mode**)
- Specify which type of client-side status to use (**stcmd set-personal-options**)
- Update the status of files (**stcmd update-status**)

Use **-?** to display the options available with the command.

For example, **stcmd -?** displays the general syntax for stcmd and **stcmd add -?** displays the options available when adding files. (Using stcmd and stcmd add without the -? also works.)

Syntax

The syntax for the command line uses the following conventions:

Table 22.1 Syntax for the Command Line

Convention	Description
Bold	Information that you must use exactly as shown.
Italics	Information that you replace with the names of your files, child folders, etc.
[ ]	Square brackets identify optional syntax.
|	A vertical bar separates mutually exclusive choices. Select only one of the choices.

Common Options

Some options show up in all or almost all of the commands explained in this chapter. In each command, they have exactly the same meaning so they are explained in this section and not repeated later. Options that do not appear in all commands or vary in meaning from command to command are explained with the commands.

All command-line syntax is interpreted as UNIX not Windows syntax.

All options can be written with either a hyphen (-) or slash mark (/). Example: -p or /p. When this chapter indicates that a option requires quotation marks, you should use them despite the fact that the option may be accepted by your operating system without them. This is both consistent and safe.

For example, Windows operating systems require quotation marks when a space is part of the option. For example, when a revision comment is several words, it must be enclosed in quotation marks. A one-word comment would *not require* quotation marks—although it is OK to use them.

Be aware that the comments sent via the Java version of the command line may fail when there are spaces in the names of working folders and files. Depending upon the shell being used and other factors, UNIX names that contain spaces often need to be enclosed in double quotation marks.

-?	Lists the command's syntax and a description of each option. -? works with each command, although not shown in the syntax. **-**h and **-**help are synonyms for **-**?. This information is sent to stderr rather than stdout. To capture this information from the Windows 2000 command prompt, you need to use "2>" rather than ">".
-active	Indicates the active process item.
-cmp	Compresses all the data sent between the workstation and the server and decompresses it when it arrives. Without this option, no compression takes place. Compression is most useful and appropriate when the client and server communicate over a slow connection. To determine whether to use compression, a small test case may be helpful. You must consider whether the time spent compressing and uncompressing data is better than the longer transfer time of uncompressed data sent over the slow connection.
-csf	When the command maps the folder specified in the -p option to the underlying folder, the use of csf causes the command to differentiate folders based on the case-sensitive spelling of their names. For example, with this option, folders named doc and Doc are recognized as different folders. Without this option, either of these folders could be recognized as the "doc" folder. The default is that folders are not differentiated based on the case of letters in their names. This option has nothing to do with case-sensitivity of file names.
-encrypt	Encrypts all the data sent between the workstation and the server and unencrypts it when it arrives. Without this option, no encryption takes place. Encryption protects files and other project information from being read by unauthorized parties over unsecured network lines. The full syntax is: **`-encrypt`** *`encryptionType`* The types of encryption are: **RC4** RSA R4 stream cipher (fast) **RC2_ECB** RSA R2 block cipher (Electronic Codebook) **RC2_CBC** RSA R2 block cipher (Cipher Block Chaining) **RC2_CFB**(Windows platforms only) RSA R2 block cipher (Cipher Feedback) These encryption types are ordered from fastest to slowest. Each of the slower encryption types is safer than the one preceding it.
eol [on|off]	Perform end-of-line conversion of text files.
-f NCI	Apply command to all files needing check in.

files...

Specifies the files to be used in the command by name or by file name-pattern specification (such as "*.c"). All options are interpreted using the semantic conventions of UNIX instead of Windows because UNIX's conventions are more specific. This means that "*", rather than "*.*" means "all files." The pattern "*.*" means "all files with file name extensions." For example, "star*.*" finds starteam.doc and starteam.cpp but not starteam. To find all of these, you could use "star*".

Without this option, the default is "*". When used, this option must always be the last option. Any options after it are ignored.

If you use * rather than "*" to indicate all files, a UNIX shell expands it into a series of items and passes this series as a group of options to the stcmd command. This can cause problems, for example, when you are checking out missing files, so it is best to use "*" to avoid unwanted complications.

If you use a set of file patterns, each pattern should be enclosed in its own set of quotation marks. For example, you can use "*.bat" "*.c", but you cannot use "*.bat *.c".

- **Note:** We recommend that you enclose this option in quotation marks, regardless of platform, but for different reasons. On Windows platforms, file and folder names that contain spaces will not be interpreted correctly unless you use quotation marks. On UNIX platforms, if you do not use quotation marks, the shell will expand the option, then pass the list of items produced by the expansion to the client. Frequently this produces unintended results. You can avoid both of these consequences by enclosing the option in quotation marks. Only if it is essential that the option be expanded by the UNIX shell is it advisable to omit the quotation marks. Mysterious failures of batch scripts on either platform may be due to this omission.

Several special characters can be used in the file specification:

*****

Matches any string including the empty string. For example, "*" matches any file name, with or without an extension. "xyz*" will match "xyz" and "xyz.cpp" and "xyzutyfj".

?

Matches any single character. For example, "a?c" will match "abc" but NOT "ac"

[...]

Matches any one of the characters enclosed by the left and right brackets.

A pair of characters separated by a hyphen (-) specifies a range of characters to be matched.
If the first character following the right bracket ([ ) is an exclamation point (!) or a caret (^), the rest of the characters are not matched. Any character not enclosed in the brackets is matched. For example, "x[a-d]y" matches "xby" but not "xey". "x[!a-d]y" matches "xey" but not "xby".

A hyphen (-) or right bracket ( ]) may be matched by including it as the first or last character in the bracketed set.

To use an asterisk (*), question mark (?), or left bracket ([ ) in a pattern you must precede it with the escape character (which is the backslash \ ).

-filter

File status filter. Statuses are C = Current, M = Modified, O = Out of Date, N = Not in View, I = Missing, G = Merge, and U = Unknown. For example, using CM applies a command only to files with a status of Current or Modified.

-fp	Overrides the specified folder's working folder or working directory. This is equivalent to setting an alternate working path for the folder. While this enables you to use a different working folder than the one specified by the folder, *its critical importance is its use to provide cross-platform compatibility*. For example, UNIX and Windows systems specify drive and directory path names in incompatible ways. Although the path "D:\MYPRODUCT\DEVELOPMENT\SOURCE" is understood on a Windows platform, it is *not* understood on a UNIX platform. Use this option to define the working path if your platform does not understand the path specified in the project. A backslash (\) is interpreted as an escape character when it precedes quotation marks. As a result, an error occurs in the following example: stcmd ci -p "xxx" -fp "C:\" "*" which is interpreted as: stcmd ci -p "xxx" -fp "C:" *" To avoid a situation like this, escape the final character in `"C:\"` as follows: stcmd ci -p "xxx" -fp "C:\\" "*" Or avoid it as follows when the -fp path doesn't end with the root folder as in "C:\orion\": stcmd ci -p "xxx" -fp "C:\orion" "*" The full syntax is: **`-fp "`***`folderPath`***`"`** Folder is the Windows term and appears in the user interface. Directory is the correct term for the UNIX platform.
-i	Prompts user to confirm check-in when file status is Merge, Out of Date, or Unknown.
-is	Applies the command to recursively to all child folders. Without this option, the command applies only to the specified folder.
-l	Locks a file.
-mark	Marks a change request as fixed, a requirement as complete, or a task as finished.
-nel	Non-exclusively locks a file
-nologo	Suppresses the copyright notice. -nologo works with each command, although not shown in the syntax.
-nomove	Do not move labels if already attached.
-o	Forces check-in.

-p

Indicates what view or folder is to be used; also provides the user name and password needed to access the server.

The full syntax is:
-p "*userName*:*password*@*hostName*:*portNumber*/*projectName*/[*viewName*/][*folderHierarchy*/]**"**

If the user name is omitted, the current user name is used.

If the password is omitted, the user is prompted to enter the password. When the user types a password, the characters are not displayed on the screen.

Entering an endpoint is required. The default port number, 49201, is used in the example.

The project name is always required.

Use a view hierarchy to identify the view. Use the colon (:) as a delimiter between view names. The view hierarchy should *always* include the root view. For example, "StarDraw:Release 4:Service Packs" indicates that the view to be used is the Service Packs view, which is a child of the Release 4 view and a grandchild of the StarDraw root view. If the view name is omitted, the root (default) view is used. If the view is the only view in that project with that name, you can use only the view name. (This is not recommended because another view with that name could be created at a later date and cause confusion.) The view name in the example is StarDraw. Because this is the root view of the StarDraw project, it could have been omitted.

Use a folder hierarchy to identify the folder. Use the forward slash (/) as a delimiter between folder names. The folder hierarchy *never* includes the root folder. Omit the folder hierarchy if the file is in the view's root folder. For example, if the root folder of the view is StarDraw and the hierarchy to your files is "StarDraw/SourceCode/Client", use only "SourceCode/Client".

If any of the variables used with this option contain characters that are used as delimiters, use the percent sign (%) followed by the hex code for each of those characters. For example, if "@" appears as a character in a password, you must replace it with "%40".

For ":", use "%3a".

For "/", use "%2f".

For "@", use "%40".

For "%", use "%25

In UNIX and other operating systems, some special characters must be preceded by a backslash "\" or another escape character. In the -p option, you can replace such characters with hex codes. For example, "%3c" could be used in UNIX instead of "\<".

For a space, use "%20".

For "<", use "%3c".

For ">", use "%3e".

-pwdfile

Specifies the path to a file that stores the user's password. This option overrides the password used as part of the -p option. It prevents others from seeing the user's password on the command line. It must be saved in UTF-8 format.

The full syntax is:
-pwdfile "*filePath***"**

The password file should consist only of the password. Leading and trailing whitespace is ignored.

-q

Suppresses progress reporting. Without this option, messages about each action appear on the screen as the action is performed.

-r

Reason for check-in.

-rf

Precedes name of file that contains the reason for the check-in.

-ro

Sets file as read-only after operation.

-rp

Specifies or overrides the working folder or working directory for the view's root folder. The stcmd add-project command uses this option to specify the new view's root folder's working folder. See for additional information.

The full syntax is:
-rp "*folderPath***"**

-rw	Sets file as read-write after operation.
-stop	Often used with -x. Halts execution of the command-line when the first error is encountered. Without this option, execution continues despite errors.
-u	Unlocks a file
-v	Version label.
-x	Switches between interactive and batch modes. Without this option, you must confirm error messages interactively and the exit codes may not be available. With this option, no error messages are displayed, but exit codes are still set. The exit codes are 0 for success and 1 for failure. For more information about exit codes, see *"Exit Codes" on page 291* and the -e option in "Comparing File Revisions: stcmd diff" on page 305.

Exit Codes

The commands in this chapter return exit codes. The codes are:

- 0 for success
- 1 for failure
- 101 if at least one of the specified file patterns did not match
- 102 if none of the specified file patterns matched

The stcmd diff command has an additional option (-e) that returns exit codes. The -e option has three exit codes (0, 1, and 2) with different meanings than those listed above. In addition, the 1 might not be returned if you do not also use the -x option.

For Windows platforms, you can use ERRORLEVEL in a batch file to perform operations based on the result of a command. For example, after an stcmd command in a batch file, you might use the following:

```
IF ERRORLEVEL int statement
```

where *int* is 0 or 1.

For example:

```
IF NOT ERRORLEVEL 1 GOTO OPOK
ECHO ERROR OCCURRED AT STEP5>LOGFILE.TXT.

:OPOK
```

For Windows NT or 2000, you can also use the pseudo environment variable %ERRORLEVEL%. For example, you might use the following in a shell or at the command line (after an stcmd command):

```
SET /A STEPNUMBER=5
SET /A THISERROR=STEPNUMBER*ERRORLEVEL
SET /A ERRORMASK=+THISERROR
```

For UNIX, each shell has its own method of evaluating exit codes. For example, in the Bourne shell, the following statement might come after an stcmd command:

```
if [ return ]; then statement
```

Adding Files: stcmd add

Use stcmd add to add files to a view from the command line.

You can simultaneously link the added files to a process item. All the files successfully added using this command will be linked and pinned to the tip revision of the process item. Use the -active option to specify the currently active process item (previously set using a client on your workstation). If no item is active or you prefer to use another item, use the option that indicates the type of the process item (-cr, -req, or -task) followed by the complete path from the root folder of the project view to the item to be used. Use the -mark option to simultaneously mark the process item as fixed, finished, or complete, depending on its type. For example, a change request can be marked fixed. The item is not marked as fixed, finished, or complete unless all the files are successfully added.

Syntax

stcmd add	**-p "***projectSpecifier***"** [**-pwdfile "***filePath***"**] [**-cmp**] [**-csf**] [**-encrypt** *encryptionType*] [**-is**] [**-q**] [**-x**] [**-stop**] [**-rp "***folderPath***"** | **-fp "***folderPath***"**] [**-l** | **-u** | **-nel** [**-ro | -fw**] [**-d "***description***"**] [**-vl "***labelName***"**] [**-eol** [**on** | **off**]] [[**-active** | [**-cr** | **-req** | **-task**] *processItemPath*] [**-mark**]] [*files...*]
-active	The active process item.
-cr	Complete path from the project view's root folder to the CR number to be used as a process item.
-d	Sets the description of the newly added files to the specified description. The description is enclosed in double quotation marks.
-eol	Automatically converts end-of-line markers to carriage return/line feeds (CR/LF) as working text files are transferred to the Server's repository. When on, the working file's current end-of-line marker is converted to a CR/LF combination. When off, the default, no end-of-line conversion is performed. You would set this option to on, for example, when you add a working file to the repository from a computer running UNIX if you want the repository to store text files as CR/LF.
-l	Locks each file after it has been added to the view. Without -l , -u, or -nel, the files are unlocked by default.
-mark	Indicates that, if all the files are successfully added, the process item's status will be changed to fixed (for a change request), finished (for a task), or complete (for a requirement). The files are pinned to the revision with the new status.
-nel	Non-exclusively locks each file after it has been added.
-req	Complete path from the project view's root folder to the Requirement number to be used as a process item.
-ro	Makes the working file read-only after this operation. Without this option, the file remains as it was prior to the operation. Usually, you use -ro to prevent yourself from editing a file that is not locked by you. -ro must be used with -l or -u or -nel. If you use -ro, you cannot use -rw.
-rw	Makes the working file read-write after this operation. Without this option, the file remains as it was prior to the operation. -rw must be used with -l or -u or -nel. If you use -rw, you cannot use -ro.
-task	Complete path from the project view's root folder to the task number to be used as a process item.

-u	Leaves the newly added files unlocked.
-vl	Specifies a label to be applied to the new files. The label is enclosed in double quotation marks. This option can appear in the command more than once. The label can be either a view or revision label, but it must already exist in the application. See also "Creating Labels: stcmd label" on page 309.

Example

The following example uses stcmd add to add the .doc files with the status Not In View to User Manual, a child of the root folder StarDraw (in the StarDraw view of the StarDraw project). It locks the files and gives them the description "First draft of chapter".

```
stcmd add -p "JMarsh:password@Orion:49201/StarDraw/ StarDraw/User Manual" -l -d
"First draft of chapter" "*.doc"
```

Adding Folders: stcmd add-folder

Use stcmd add-folder to add folders to a view from the command line. You can add it to the root folder or any other folder in that view.

The working folder for your new folder is created by default within the application—not on your workstation. The working folder has the same name as the folder. It is a child folder of the folder's parent's working folder.

For example, suppose you create a folder named "Wizard". Wizard is a child of a folder whose working folder is "C:\StarDraw". Wizard's working folder becomes "C:\StarDraw\Wizard".

Using -is enables you to add a branch of folders to the project view's folder hierarchy. When you use -is, use either -rp or -fp to specify the folder on your workstation whose child folders will become the new folder's child folders. Using -fp is recommended, as it specifies the path directly to the parent of those child folders. This is because -rp, which specifies the path to the view's root folder's working folder, appends the folder names in the hierarchy from the root folder to the new folder to the path you specify. Only when you use -is do -rp and -fp have any effect on this command.

Syntax

stcmd add-folder	**-p** "projectSpecifier" [**-pwdfile** "filePath"] [**-cmp**] [**-csf**] [**-encrypt** encryptionType] [-is] [**-q**] [**-x**] [**-stop**] [**-rp** "folderPath" | **-fp** "folderPath"] **-name** "folderName" [**-d** "description"] [**-ex** "excludeType"] [**-exlist** "fileMask" | **-exfile** "filePath"]
-d	Specifies a description for the folder. Use a maximum of 254 characters.

-ex	Indicates the exclude lists to be used by this new folder. Exclude lists exclude certain files or types of files from visibility. If a working file in this folder's working folder would have the status Not In View but it matches a file specification in one of the exclude lists, the application does not display it at all. It is as though the file did not exist. For example, suppose you are creating files in an application that makes automatic backup copies of each file (with the extension .bak) every time you save a file. Your working folder might contain several .bak files, but you have no reason to add them to the project view. From the application, it is annoying to see these .bak files as possible candidates, so you exclude them. Excluding files is done on a per-folder basis. However, exclude lists can be inherited from parent folders. The full syntax is: **-ex** *excludeType* The types are: **inherit**, which indicates that this folder will inherit any exclude lists used by its parent folder and use the exclude list specified with either **-exfile** or **-exlist** (if one is created). This is the default. **local**, which Indicates that this folder will use only the exclude list specified with either **-exfile** or **-exlist**. **none,** which indicates that this folder will use no exclude lists, regardless of what you specify with either **-exfile** or **-exlist**.
-exfile	Specifies the path to the file that contains the local exclude list for this folder. See -exlist for a description of the exclude list's contents.
-exlist	Specifies the local exclude list for this folder. Use a maximum of 254 characters. Enter one or more file specifications (using the standard * and ? wild cards), separated by commas, spaces, or semicolons. To include a comma, space, or semicolon as part of the specification, enclose the specification in double quotation marks. For example: `*.exe,*.dll p*z.doc;*.t?t "test *.*"` If you are using double-quotation marks in your exclude list or have a lengthy exclude list, we recommend that you use the -exfile option. With -exlist, each quotation mark in the exclude list needs to be preceded by the escape character for your system or shell. For example, the caret (^) works on NT systems. With -exfile, you do not need to use escape characters.
-name	Specifies a name for the folder. Use a maximum of 254 characters. In a file, if the exclude list contains double quotation marks, the escape character is unnecessary.

Example

The following example uses stcmd add-folder to create a folder named Wizard as a child of the StarDraw folder, the root folder of the StarDraw project view. In addition, it sets a local exclude list for Wizard. By default, Wizard inherits its parent folder's exclude lists and use the local one as well.

```
stcmd add-folder -p "JMarsh:password@Orion:49201/ StarDraw/StarDraw/" -name
"Wizard" -d "StarDraw setup wizard" -exlist "*.bak"
```

The next example creates the same folder as in the previous example. However it includes child folders. In this case, the folder with the path "C:\Wizard" has child folders (Source, Spec, and Doc), all of which are added as folders in addition to Wizard. All of the new folders (Wizard, Source, Spec, and Doc) will have the default working folders assigned to them automatically by the Server, regardless of the setting for -fp. Wizard will be the parent of Source, Spec, and Doc. StarDraw is the parent of Wizard.

```
stcmd add-folder -p "JMarsh:password@Orion:49201/ StarDraw/StarDraw/" -name
"Wizard" -d "StarDraw setup wizard" -is -fp "C:\" -exlist "*.bak"
```

Adding Projects: stcmd add-project

Use stcmd add-project to add a project to a Server configuration from the command line. When a project is created, so is its root view and the root view's root folder. In this command, the -rp option specifies the working folder for that root folder.

Using -is enables you to use the working folder's child folders as the root folder's child folders in the folder hierarchy.

Syntax

stcmd add-project[	**-pwdfile** "*filePath*"] [**-cmp**] [**-encrypt** *encryptionType*] [-**is**] [**-q**] [**-x**] [**-stop**] **-s** "*serverName*" **-name** "*projectName*" **-rp** "*folderPath*" [**-d** "*description*"] [**-kw** "*fileMask*" | **-kwfile** "*filePath*"] [-**ex** "*excludeType*"] [**-exlist** "*fileMask*" | -**exfile** "*filePath*"]
-d	Specifies a description for the project. Use a maximum of 254 characters.
-ex	Indicates the exclude lists to be used by the project's root folder. Exclude lists exclude certain files or types of files from visibility. If a working file in this folder's working folder would have the status Not In View but it matches a file specification in one of the exclude lists, the application does not display it at all. It is as though the file did not exist. For example, suppose you are creating files in an application that makes automatic backup copies of each file (with the extension .bak) every time you save a file. Your working folder might contain several .bak files, but you have no reason to add them to the project view. From the application, it is annoying to see these .bak files as possible candidates, so you exclude them. Excluding files is done on a per-folder basis. However, exclude lists can be inherited from parent folders The full syntax is: **-ex** *excludeType* The types are: **inherit**Indicates that the root folder will inherit any exclude lists used by its parent folder and use the exclude list specified with either **-exfile** or **-exlist** (if one is created). This is the default—even though the root folder has nothing to inherit. **local**Indicates that the root folder will use only the exclude list specified with either **-exfile** or **-exlist**. **none**Indicates that the root folder will use no exclude lists, regardless of what you specify with either **-exfile** or **-exlist**.
-exfile	Specifies the path to the file that contains the local exclude list for the root folder. See -exlist for a description of the exclude list's contents.

-exlist	Specifies the local exclude list for the root folder. Use a maximum of 254 characters. Enter one or more file specifications (using the standard * and ? wild cards), separated by commas, spaces, or semicolons. To include a comma, space, or semicolon as part of the specification, enclose the specification in double quotation marks. For example: `*.exe,*.dll p*z.doc;*.t?t "test *.*"` If you are using double-quotation marks in your exclude list or have a lengthy exclude list, we recommend that you use the -exfile option. With -exlist, each quotation mark in the exclude list needs to be preceded by the escape character for your system or shell. For example, the caret (^) works on NT systems. With -exfile, you do not need to use escape characters.
-kw	Specifies the file extensions with which you want to use keywords. Use a maximum of 254 characters. Enter one or more file specifications (using the standard * and ? wild cards), separated by commas, spaces, or semicolons. To include a comma, space, or semicolon as part of the specification, enclose the specification in double quotation marks. For example: `*.cpp,*.h p*z.doc;*.t?t "test *.*"` If you are using double-quotation marks in your keyword list or have a lengthy list, we recommend that you use the -kwfile option. With -kwlist, each quotation mark in the keyword list needs to be preceded by the escape character for your system or shell. For example, the caret (^) works on NT systems. With -kwfile, you do not need to use escape characters.
-kwfile	Specifies the path to the file containing the file extensions with which you want to use keywords.
-name	Specifies a name for the project. Use a maximum of 254 characters.
-s	Identifies the Server. The full syntax is: **-s "***userName***:***password@host***:***portNumber***"** For example: -s "JMarsh:password@orion:49201" If the user name is omitted, the current user name is used. The user name in the example is “JMarsh” If the password is omitted, the user is prompted to enter the password. The password in the example is “password” If the host name is omitted, the default is localhost. The host name in the example is “orion”. The port number is required. The default port number, 49201, is used in the example.

Example

The following example uses stcmd add-project to create a project named Integrations on the computer named Orion. (Orion is running an instance of the Server with a server configuration that uses port 49201.) This command creates the project, specifies that the data sent between workstations and the server will be compressed and encrypted, and gives the project a description.

```
stcmd add-project -s "JMarsh:password@Orion:49201" -cmp -encrypt "RC4" -name
"Integrations" -rp "C:\integrations" -d "integrations between our products and
our partner’s products"
```

Adding Views: stcmd add-view

Use stcmd add-view to add a view to a Server configuration from the command line. When the view is created, its parent view is the view specified with the -p option and its root folder is the folder specified with the -p option. In this command, the -rp option specifies the working folder for the root folder.

Use the following options to create the following types of views:

- Use -dr to create a read/write reference view.
- Use -dr -ro to create a read-only reference view.
- Use -dr -ba to create a branching view in which the behavior of existing items is set to branch on change.
- Use -dr -bn to create a branching view in which the behavior of existing items is *not* set to branch on change.
- If you do *not* use -dr, a blank view is created.

If you use -ro, -ba, or -bn without the -dr option, an error message is displayed.

If you use -cfgl, -cfgp, or -cfgd without both the -dr option and one of the -ro, -ba, or -bn options, an error message is displayed.

See the *StarTeam Administator's Guide* for more information about each type of view.

Syntax

stcmd add-view	**p** "projectSpecifier" [**-pwdfile** "*filePath*"] [-cmp] [-**encrypt** *encryptionType*] [**-q**] [**-x**] [**-stop**] **-name** "*viewName*" [**-rp** "*folderPath*"] [-d "*description*"] [**-d**r [**-ro** |**-ba** |**-bn** [-**cfgl** "*labelName*"| **-cfgp** "*stateName*"	**-cfgd** "*asOfDate*"]]]
-ba	When used with -dr, specifies a branching view in which the behavior of existing items is set to branch on change. The value of the view property Set Items Shared Into View To Branch On Change is initially set. This option must be used with -dr.
-bn	When used with -dr, specifies a branching view in which the behavior of existing items is *not* set to branch on change. The value of the view property Set Items Shared Into View To Branch On Change is initially cleared. This option must be used with -dr.
-cfgd	Configures the view as of the specified date/time. Examples include: ▪ "12/29/01 10:52 AM" ▪ "December 29, 2001 10:52:00 AM PST" ▪ "Monday, December 29, 2001 10:52:00 oclock AM PST" This option must be used with one of the following combinations: -dr -ro, -dr -ba, or -dr -bn.
-cfgl	Configures the view using the specified label. Without -cfgl or -cfgp or -cfgd, the view's current configuration is used. This option must be used with one of the following combinations: -dr -ro, -dr -ba, or -dr -bn.
-cfgp	Configures the view using the specified promotion state. This option must be used with one of the following combinations: -dr -ro, -dr -ba, or -dr -bn.

-d	Specifies a description for the view. Use a maximum of 254 characters.
-dr	Specifies a derived view. All views, except blank views are derived. See also -ba, -bn, and -ro. When used without -ba, -bn, or -ro, a read/write reference view is created. The configuration of a read/write reference view is the same configuration as its parent view. Therefore, using -dr without -ba, -bn, or -ro, but with -cfgl, -cfgp, or -cfgd results in an error message. When this option is not used, a blank view is created. For blank views, the value of the view property named Set Items Shared Into View To Branch On Change is initially cleared.
-name	Specifies a name for the view. Use a maximum of 254 characters.
-ro	When used with -dr, specifies a read-only reference view.

Examples

The following example uses stcmd add-view to create a branching view named Maintenance 5.1 on the computer named Orion. (Orion is running an instance of the Server with a server configuration that uses port 49201.)

This command creates the view as a child of the existing StarDraw view and uses the StarDraw folder as its root folder. The new view is based on the label used for the last build of the 5.1 product before it shipped (Build 403). It has a working folder that is different from the parent's working folder. All the existing items in the view will have their behavior set to branch on change.

```
stcmd add-view -p "JMarsh:password@Orion:49201/ StarDraw/StarDraw/" -cmp
-encrypt "RC4" -name "Maintenance 5.1" -rp "C:\StarDraw\Maintenance 5.1" -d
"Maintenance view for 5.1 release of our product" -dr -ba -cfgl "Build 403"
```

The following example uses stcmd add-view to create a read/write reference view named Rooted At Source Code on the computer named Orion. (Orion is running an instance of the Server with a server configuration that uses port 49201.)

This command creates the view as a child of the existing StarDraw view and uses the SourceCode folder as its root folder. It has the same working folder as its parent.

Because a read/write reference view has to have the same configuration as its parent, none of the -cfgl, -cfgp, and -cfgl options cannot be used.

```
stcmd add-view -p "JMarsh:password@Orion:49201/ StarDraw/StarDraw/Source Code"
-cmp -encrypt "RC4" -name "Rooted At SourceCode" -d "StarDraw main view but with
SourceCode folder as the root of the hierarchy" -dr
```

Applying Labels: stcmd apply-label

Use stcmd apply-label to label specified file revisions with view or revision labels. The labels must already exist in the application. You can create the labels in the application or with the stcmd label command. See "Creating Labels: stcmd label" on page 309.

Syntax

stcmd apply-label	**-p** "*projectSpecifier*" [**-pwdfile** "*filePath*"] [**-cmp**] [**-csf**] [**-encrypt** *encryptionType*] [**-is**] [**-q**] [**-x**] [**-stop**] [**-rp** "*folderPath*" | **-fp** "*folderPath*"] [**-filter** "*fileStatus*"] [**-vl** "*labelName*" | **-vd** "*asOfDate*" | **-vn** *revisionNumber*] **-lbl** "*labelName*" [*files...*]
-filter	Specifies a string of one or more characters, each of which represents a file status. Never include spaces or other whitespace in this string. The label is applied only to the files that currently have the specified statuses. You cannot apply labels to files that are Not In View. The letters used to represent the statuses are: C for Current, M for Modified, G for Merge, O for Out of Date, I for Missing, and U for Unknown.
-lbl	Specifies the label name to be added to the specified revisions. This option can be used more than once. The application attaches all the labels to the specified file or revisions.
-vd	Specifies the as-of date/time used to identify the revisions that get the new label. Examples include: ▪ "12/29/01 12:41 PM" ▪ "December 29, 2001 12:41:21 PM" ▪ "Monday, December 29, 2001 12:41"
-vl	Specifies the revision or view label used to identify the revisions that get the new label. This label must already exist in the application. Without the -vn or -vd or -vl option, the tip revision of each file receives the label.
-vn	Specifies the revision number used to identify the revisions that get the new label.

Example

The following example uses stcmd apply-label to apply the label Beta to files in User Manual, a child of the root folder StarDraw (in the StarDraw view of the StarDraw project). The application applies the label to the revisions of those files that were current at noon on July 7, 2003.

```
stcmd apply-label -p "JMarsh:password@Orion:49201/StarDraw/StarDraw/User Manual"
-vd "07/07/03 12:00 PM" -lbl "Beta" "*"
```

Checking in Files: stcmd ci

Use stcmd ci to check files in to an application repository from your working folder using the command line.

You can simultaneously link the new file revisions to a process item. All the files successfully added using this command will be linked and pinned to the tip revision of the process item. Use the -active option to specify the currently active process item (previously set using an application client on your workstation). If no item is active or you prefer to use another item, use the option that indicates the type of the process item (-cr, -req, or -task) followed by the complete path from the root folder of the project view to the item to be used. Use the -mark option to simultaneously mark the process item as fixed, finished, or complete, depending on its type. For example, a change request can be marked fixed. The item is not marked as fixed, finished, or complete unless all the files are successfully added.

Syntax

stcmd ci	**-p** "*projectSpecifier*" [**-pwdfile** "*filePath*"] [**-cmp**] [**-csf**] [**-encrypt** *encryptionType*] [**-is**] [**-q**] [**-x**] [**-stop**] [**-rp** "*folderPath*" | **-fp** "*folderPath*"] [**-filter** "*fileStatus*"] [**-l** | **-u** | **-nel** [**-ro** | **-rw**]] [**-vl** "*labelName*"] [**-nomove**] [**-f NCI**] [**-o** | **-i**] [**-r** "*comment*" | **-rf** "*filePath*"] [**-eol** [**on** | **off**]] [**-active** | [**-cr** | **-req** | **-task**] *processItemPath* [**-mark**]] [*files...*]
-active	The active process item.
-cr	Complete path from the project view's root folder to the CR number to be used as a process item.
-eol	Automatically converts end-of-line markers to carriage return/line feeds (CR/LF) as working text files are transferred to the Server's repository. When on, the working file's current end-of-line marker is converted to a carriage return/line feed (CR/LF) combination. When off, the default, no end-of-line conversion is performed. For Windows clients, the end-of-line marker is CR/LF; for UNIX platforms, it is a line feed (LF). You would set this option to on, for example, when you check a working file into the repository from a computer running UNIX and the repository stores text files as CR/LF.
-f NCI	Specifies the check-in of any file whose status is Modified. NCI stands for "needs check-in." No other types of files are selected for check-in. -f NCI is ignored if -filter is used.
-filter	Specifies a string of one or more characters, each of which represents a file status. Never include spaces or other whitespace in this string. Only files that currently have the specified statuses will be checked in. You cannot check in files that are Not In View. The letters used to represent the statuses are: C for Current, M for Modified, G for Merge, O for Out of Date, I for Missing, and U for Unknown. -filter takes precedence over -f NCI. If you use G, O, or U, you must also specify -i or -o. Otherwise, the G, O, or U is ignored.
-i	Allows an interactive check-in for files whose status would normally not allow them to be checked in. You are asked about each file whose status is Merge, Out of Date or Unknown. You can force the file to be checked in with your response. If you use the -i option, you cannot use the -o option.
-l	Locks each file after it has been checked in. Without -l, -u, or -nel, the files lock status is unchanged.
-mark	Indicates that, if all the files are successfully added, the process item's status will be changed to fixed (for a change request), finished (for a task), or complete (for a requirement). The files are pinned to the revision with the new status.
-nel	Non-exclusively locks each file after it has been checked in.
-nomove	Stops the application of the label specified by the -vl option if the file, which is being checked in, already has a revision with that label. Otherwise, the label will be moved from the currently labeled revision to the newly checked-in revision.
-o	Forces check-in for files whose status would normally not allow them to be checked in. This option forces all files whose status is Merge, Out of Date or Unknown to be checked in. If you use the -o option, you cannot use the -i option.

-r	Provides a revision comment, usually the reason for checking in the files. If you use the -r option, you cannot use the -rf option.
-req	Complete path from the project view's root folder to the Requirement number to be used as a process item.
-rf	Provides the path to the file that contains the revision comment. If you use the -rf option, you cannot use the -r option.
-ro	Makes the working file read-only after this operation. Without this option, the file remains as it was prior to the operation. Usually, you use -ro to prevent yourself from editing a file that is not locked by you. -ro must be used with -l or -u or -nel. If you use -ro, you cannot use -rw.
-rw	Makes the working file read-write after this operation. Without this option, the file remains as it was prior to the operation. -rw must be used with -l or -u or -nel. If you use -rw, you cannot use -ro.
-task	Complete path from the project view's root folder to the task number to be used as a process item.
-u	Unlocks the newly checked-in files.
-vl	Specifies a label (created using stcmd label) to be applied to the checked-in files. The label is enclosed in double quotation marks. This option can appear in the command more than once. The label can be either a view or revision label, but it must already exist in the application. See also "Creating Labels: stcmd label" on page 309.

Example

The following example uses stcmd ci to check in .bmp files to Online Help, a child of the root folder StarDraw (in the StarDraw view of the StarDraw project). The command unlocks the files, makes the working copy read-only, and provides a revision comment (a reason for checking in the files).

```
stcmd ci -p "JMarsh:password@Orion:49201/StarDraw/StarDraw/SourceCode/Online
Help" -u -ro
-r "revised for beta" "*.bmp"
```

Checking Out Files: stcmd co

Use stcmd co to check out files from an application repository to your working folder using the command line. This command attempts to check out all the files (or all the specified files) from the specified location, regardless of their statuses. Unless you use -o, this command pauses at each file with a Modified, Merge or Unknown status to let you know that the file will not be checked out.

With the -merge option, you can merge files as part of the check-out process. Merging is not part of the check-in process.

Syntax

stcmd co

```
 -p "projectSpecifier" [-pwdfile "filePath"]
[-cmp] [-csf] [-encrypt encryptionType]
[-is] [-q] [-x] [-stop]
[-rp "folderPath" | -fp "folderPath" ]
[-cfgl "labelName"| -cfgp "stateName"
| -cfgd "asOfDate"]
[-filter "fileStatus"] [-o | -i |
-merge [ reportOption ][-hook "appName"]]
[-l | -u | -nel [-ro | -rw]]
[-vl "labelName" | -vd "asOfDate" |
-vn revisionNumber] [-f NCO] [-ts]
[-eol on | off | cr| lf| crlf ] [-fs] [files...]
```

-cfgd

Configures the view as of the specified date/time. Examples include:

```
"12/29/01 10:52 AM"
"December 29, 2001 10:52:00 AM PST"
"Monday, December 29, 2001 10:52:00 oclock AM PST"
```

-cfgl

Configures the view using the specified label. Without -cfgl or -cfgp or -cfgd, the view's current configuration is used.

-cfgp

Configures the view using the specified promotion state.

-eol

Can automatically convert end-of-line markers.

When on, text files are transferred from the Server's repository to the workstation's working folder with the end-of-line convention for the platform executing the command as determined by the Java VM.

When off, the default, no end-of-line conversion is performed. Using off is the same as not using -eol at all.

When you specify the end-of-line character (cr, lf, or crlf), text files are transferred from the Server's repository to the workstation's working folder with the specified end-of-line convention.

For Windows clients, the end-of-line marker is a carriage return/line feed (crlf) combination; for UNIX platforms, it is a line feed (lf); for MAC systems, a carriage return (cr).

You would set this option to on or lf, for example, when you compare a file from the repository and a working file on a UNIX system (if the repository stores text files as crlf).

-f NCO

Specifies the check-out of any file whose status is Missing or Out of Date. NCO stands for "needs check-out." No other files are selected for check-out.

-f NCO is ignored if -filter is used.

-filter

Specifies a string of one or more characters, each of which represents a file status. Never include spaces or other whitespace in this string. Only files that currently have the specified statuses will be checked out. You cannot check out files that are Not In View. The letters used to represent the statuses are:
C for Current, M for Modified, G for Merge, O for Out of Date, I for Missing, and U for Unknown.

-filter takes precedence over -f NCO. If you use G, M, O, or U, you must also specify -merge or -o to force the check-out operation. Otherwise, the G, M, O, or U is ignored.

-fs

Prevents file statuses from being remembered after the check-out occurs. Subsequent status values for these files will be incorrect and indeterminate. Use this option where a file's status is irrelevant. For example, if you routinely delete the working folders before checking out files for a build, there are no files and their statuses do not matter.

Be aware that the file statuses may never be known—even if you use the update-status command later. You can do a force check out without the -fs option to obtain current files with correct statuses.

-hook

Used only with -merge. Enables you to specify an alternate application (other than that available with the application) to perform the merge.

The value of the option should be the name of a program to run to perform the merge, for example:

```
-hook mymerge.sh
```

The merge application must return an exit code of 0 meaning that no conflicts were detected and an exit code of 1 meaning that conflicts were detected. Any other value indicates an error.

The merge application cannot be a batch file on Windows platforms because, when run via Java, the result code is not returned properly.

For each file to be merged, stcmd provides three arguments to the merge application. The arguments are the fully qualified paths to the following three files (in this order):

localFile: The local working file

commonFile: The contents of the file revision in the repository which is the most recent common version between the file being modified locally and the specified revision in the repository

otherFile: The contents of the file revision being checked out

The merge hook program must send the merge result to standard output (stdout).

-i

Allows an interactive check-out for files whose status would normally not allow them to be checked out. You are asked about each file whose status is Modified, Merge, or Unknown. You can force the file to be checked out with your response.

If you use the -i option, you cannot use the -o or -merge option.

-l

Locks each file after it has been checked out. If -l or -u or -nel is not used, the files lock status remains unchanged.

-merge

Enables you to merge the working file with the revision being checked out and with the revision upon which both of these is based. The working file must have the status Merge.

You can specify one of the following reporting options with -merge:

-dryrun: Indicates whether the merged results file has conflicts or not; the local working file is unchanged as nothing is checked out. This provides a preview.

-alwaysprompt: Always prompts the user to save the merged result file in the working folder whether or not there were merge conflicts.

-neverprompt: Always saves the merged results file to the working folder.

-conflictprompt: Prompts the user to save the merged results file only if conflicts were detected.

The options -dryrun, -alwaysprompt, -neverprompt, -conflictprompt are mutually exclusive. When none are specified, the default behavior is -conflictprompt.

You can specify an alternate application (other than that available with the application) to perform the merge. See “-hook” on page 303.

If you use the -merge option, you cannot use the -i or -o option.

If you are not using -hook and you save a merged file with conflicts, each conflict is marked inside the working file as follows:

```
<<<<<<< fileName (local)
line as it appears in the local file
=======
line as it appears in the revision being checked out
>>>>>>> fileName (version x)
```

where *fileName* is the name of the file and *x* is the number of the revision being checked-out.

If you do not specify -merge, files with the status Merge are ignored unless you specify -o and force the check-out operation.

-nel	Non-exclusively locks the file after it has been checked out.
-o	Forces the check-out of files of any status. If you use the -o option, you cannot use the -i or -merge option.
-ro	Makes the working file read-only after this operation. Without this option, the file remains as it was prior to the operation. Usually, you use -ro to prevent yourself from editing a file that is not locked by you. -ro must be used with -l or -u or -nel. If you use -ro, you cannot use -rw.
-rw	Makes the working file read-write after this operation. Without this option, the file remains as it was prior to the operation. -rw must be used with -l or -u or -nel. If you use -rw, you cannot use -ro.
-ts	Sets each working file's time stamp to the check-out time. Without this option, the file is given the same time stamp as the checked-in revision of the file.
-u	Unlocks the checked-out files.
-vd	Specifies the as-of date/time used to identify the revisions to be checked out. The last revision before the specified date/time is the one checked out for each file. See the date/time examples for -cfgd above.
-vl	Specifies the revision or view label used to identify the revisions to be checked out. Without the -vn or -vd or -vl option, the tip revision of each file is checked out.
-vn	Specifies the revision number of the files to be checked out.

Example

The following example uses stcmd co to lock and check out .doc files from User Manual, a child of the root folder StarDraw (in the StarDraw view of the StarDraw project).

```
stcmd co -p "JMarsh:password@Orion:49201/StarDraw/StarDraw/User Manual" -l
"*.doc"
```

The next example uses stcmd co to merge a readme file.

```
stcmd co -p "NTesla:@10.50.5.179:49201/WebDev/WebDev" -encrypt RC4 -fp "/export/
home0/johnson/working" -merge "README"
```

Deleting Local Files: stcmd delete-local

Use stcmd delete-local to delete files from a working folder. You can delete files that are under version control as well as files that are not in the application. This does not remove any files from version control; it merely reduces the amount of data stored on your workstation in a working folder. If you are deleting files based on their status, it is a good idea to use stcmd update-status first. For more information, see "Updating File Status: stcmd update-status" on page 316.

Syntax

stcmd delete-local	**-p** "*projectSpecifier*" [**-pwdfile** "*filePath*"] [**-cmp**] [**-csf**] [**-encrypt** *encryptionType*] [**-is**] [**-q**] [**-x**] [**-stop**] [**-rp** "*folderPath*" | **-fp** "*folderPath*"] [**-cfgl** "*labelName*" | **-cfgp** "*stateName*" | **-cfgd** "*asOfDate*"] [**-filter** "*fileStatus*"] [*files...*]
-cfgd	Configures the view as of the specified date/time. Examples include: "12/29/01 10:52 AM" "December 29, 2001 10:52:00 AM PST" "Monday, December 29, 2001 10:52:00 oclock AM PST"
-cfgl	Configures the view using the specified label. Without -cfgl or -cfgp or -cfgd, the view's current configuration is used.
-cfgp	Configures the view using the specified promotion state.
-filter	Specifies a string of one or more characters, each of which represents a file status. Never include spaces or other whitespace in this string. Only files that currently have the specified statuses will be deleted. The letters used to represent the statuses are: The letters used to represent the statuses are: C for Current, M for Modified, G for Merge, O for Out of Date, I for Missing, and U for Unknown.

Example

The following example uses stcmd delete-local to delete some files from the working folder for the folder named SourceCode. SourceCode is a child of the root folder StarDraw (in the StarDraw view of the StarDraw project). It deletes all the files that are not under version control. Those files have the file status Not In View.

```
stcmd delete-local -p "JMarsh:password@Orion:49201/StarDraw/StarDraw/SourceCode"
/filter "N" "*"
```

Comparing File Revisions: stcmd diff

Use stcmd diff to display differences between two revisions of a text or a binary file. The command can specify more than one file, each of which gets a similar comparison. If you do not specify any revisions (using -vn or -vd or -vl), the working copy of each specified file is compared to its tip revision in the repository. If you specify a single revision, the working copy of each specified file is compared to that revision. If you specify two revisions, those two revisions of each specified file are compared. When comparing text files, the differences can be displays. Binary comparisons result in output that indicates whether the revisions of the file are the same or different.

Syntax

stcmd diff **-p** "*projectSpecifier*"
[**-pwdfile** "*filePath*"] [**-cmp**] [**-csf**]
[**-encrypt** *encryptionType*]
[**-is**] [**-q**] [**-x**] [**-stop**]
[**-rp** "*folderPath*" | **-fp** "*folderPath*"]
[**-cfgl** "*labelName*" |**-cfgp** "*stateName*"
| **-cfgd** "*asOfDate*"]
[**-filter** "*fileStatus*"]
[**-eol** [**on** | **off** | **cr**| **lf**| **crlf**]]]
[**-w** | **-Bpvcs** | **-b**] [**-i**] [**-m** "*maskSet*"]
[**-t** *number*] [**-c** *number*] [**-n]** [**-nd**] [**-e**]
[**-vl** "*labelName*" | **-vd** "*asOfDate*"
| **-vn** *revisionNumber*] [*files...*]

-b — When comparing two lines of text files, ignores trailing whitespace and treats all other strings of whitespace as equal in length. For example, the following lines are equivalent:

```
" hi mom   "
"    hi    mom"
```

-Bpvcs — Ignores leading and trailing whitespace when comparing two lines of text files. For example, the following lines are equivalent because there is only one space between "hi" and "mom":

```
"       hi mom   "
"     hi mom"
```

but the next line is not equivalent:

```
"hi              mom"
```

-c — Specifies the number of unchanged lines to display before and after a difference is found in text files. Without this option, all lines of the files are displayed. For example, -c 2 places two unchanged lines before and after each line or set of lines that has changed.

-cfgd — Configures the view as of the specified date/time. Examples include:

```
"12/29/01 10:52 AM"
"December 29, 2001 10:52:00 AM PST"
"Monday, December 29, 2001 10:52:00 oclock AM PST"
```

-cfgl — Configures the view using the specified label. Without -cfgl or -cfgp or -cfgd, the view's current configuration is used.

-cfgp — Configures the view using the specified promotion state.

-e — Causes the command to return the following exit codes:

- 0 if all the compared files are equivalent
- 1 if an error condition occurs
- 2 if at least one file is different

Use -e with either text or binary files.

-eol — This command is irrelevant at this point in time because diff currently ignores end-of-line markers. If two lines are the same except for this, they are reported to be identical.

-filter — Specifies a string of one or more characters, each of which represents a file status. Never include spaces or other whitespace in this string. Only files that currently have the specified statuses will be compared. You cannot compare revisions of files that are Not In View.

The letters used to represent the statuses are:
C for Current, M for Modified, G for Merge, O for Out of Date, I for Missing, and U for Unknown.

-i — Ignores the case of letters when comparing two text files. For example, "A" is equivalent to "a".

-m	When comparing two text files, ignores the characters in certain columns as specified by one or more masks. Each mask has the following syntax: "*columnNumber*-*columnNumber*[**(numeric)**]" For example, "1-6" ignores the characters in the first six columns of each line, and "1-6(numeric)" ignores the first six columns of each line if the character in column 1 is a digit in both files. You can use a series of masks, but they must be separated by commas. The syntax is: "*mask*[,*mask*]..."
-n	Suppresses the display of line numbers in the two text files.
-nd	Suppresses the display of differences in two text files. Comparisons of binary files do not display differences. This option is useful with the -e option.
-t	Specifies the number of spaces to use for each tab stop when displaying the file differences for text files. The default is four. Use -t 0 to suppress tab conversion.
-vd	Specifies the as-of date/time used to identify the revisions to be compared. The last revision before the specified date/time is the one used. See the date/time examples for -cfgd above.
-vl	Specifies the revision or view label used to identify the revisions to be compared. You can specify any combination of zero, one or two of the -vn, -vd, -vl options. Use zero options to compare the working file to the tip revision, one to compare the working file to the specified revision, and two to compare two revisions.
-vn	Specifies the revision number to be compared.
-w	Ignores all whitespace (tabs and spaces) when comparing two lines in text files. For example, the following lines would be equivalent: "a = (b + 2);" "a=(b+2);" The -w, -Bpvcs, and -b options are mutually exclusive.

Example

The following example uses stcmd diff to compare the Beta1 and Beta2 revisions of each of the .cpp files in SourceCode, a child of the root folder StarDraw (in the StarDraw view of the StarDraw project). It ignores all whitespace.

```
stcmd diff -p "JMarsh:password@Orion:49201/StarDraw/StarDraw/SourceCode" -w -vl
"Beta1" -vl "Beta2" "*.cpp"
```

Changing Descriptions: stcmd dsc

Use stcmd dsc to change a file description from the command line. This command creates a new file revision with the new description as one of its properties.

Syntax

stcmd dsc	**-p** "*projectSpecifier*" [**-pwdfile** "*filePath*"] [**-cmp**] [**-csf**] [**-encrypt** *encryptionType*] [**-is**] [**-q**] [**-x**] [**-stop**] [**-rp** "*folderPath*" | **-fp** "*folderPath*"] [**-filter** "*fileStatus*"] **-d** "*description*" [*files...*]
-d	Provides a description for the files. The description is enclosed in double quotation marks.
-filter	Specifies a string of one or more characters, each of which represents a file status. Never include spaces or other whitespace in this string. Only files that currently have the specified statuses will be given the description. You cannot change the descriptions of files that are Not In View. The letters used to represent the statuses are: C for Current, M for Modified, G for Merge, O for Out of Date, I for Missing, and U for Unknown.

Example

The following example uses stcmd dsc to change the description of stdafx.cpp in User Manual, a child of the root folder StarDraw (in the StarDraw view of the StarDraw project).

```
stcmd dsc -p "JMarsh:password@Orion:49201/StarDraw/StarDraw/SourceCode" -d
"SourceCode for StarTeam" "stdafx.cpp"
```

Displaying File History: stcmd hist

Use stcmd hist to display the revision history of files.

Syntax

stcmd hist	**-p** "*projectSpecifier*" [**-pwdfile** "*filePath*"] [**-cmp**] [**-csf**] [**-encrypt** *encryptionType*] [**-is**] [**-q**] [**-x**] [**-stop**] [**-rp** "*folderPath*" | **-fp** "*folderPath*"] [**-cfgl** "*labelName*" | **-cfgp** "*stateName*" | **-cfgd** "*asOfDate*"] [**-filter** "*fileStatus*"] [*files...*]
-cfgd	Configures the view as of the specified date/time. Examples include: "12/29/01 10:52 AM" "December 29, 2001 10:52:00 AM PST" "Monday, December 29, 2001 10:52:00 oclock AM PST"
-cfgl	Configures the view using the specified label. Without -cfgl or -cfgp or -cfgd, the view's current configuration is used.
-cfgp	Configures the view using the specified promotion state.
-filter	Specifies a string of one or more characters, each of which represents a file status. Never include spaces or other whitespace in this string. Only files that currently have the specified statuses will be reported. You cannot review the history of files that are Not In View. The letters used to represent the statuses are: C for Current, M for Modified, G for Merge, O for Out of Date, I for Missing, and U for Unknown.

Example

The following example uses stcmd hist to display the revision history of the file star.h in SourceCode, a child of the root folder StarDraw (in the StarDraw view of the StarDraw project).

```
stcmd hist -p "JMarsh:password@Orion:49201/StarDraw/StarDraw/
SourceCode" "star.h"
```

Creating Labels: stcmd label

Use stcmd label to create a view label or a revision label. If the label is a view label, it can be designated as a build label. By default, view labels are automatically applied to every folder, file, change request, topic and task in the view. By default, revision labels are not applied to any items.

You can use stcmd apply-label to apply labels created with stcmd label to the specified files. See "Applying Labels: stcmd apply-label" on page 298. You can also use the label option (-vl) in stcmd ci to attach files to your new label as you check those files in. See "Checking in Files: stcmd ci" on page 299.

Syntax

stcmd label	**-p** "*projectSpecifier*" [**-pwdfile** "*filePath*"] [**-cmp**] [**-encrypt** *encryptionType*] [**-q**] [**-x**] [**-stop**] **-nl** "*labelName*" [**-vl "***labelName***"** | **-vd "***asOfDate***"** | **-vp "***stateName***"**] [**-d "***description***"**] [**-b** | **-r**] [**-f**]
-b	Specifies that the new label is a build label. Without either -b or -r, the label is a view label. View labels (and a build label is a special type of view label) are immediately and automatically applied to every folder, file, change request, task, and topic in the view.
-d	Specifies the description of the label.
-f	Creates the new label as a frozen label.
-nl	Specifies the new label's name.
-r	Specifies that the new label is a revision label. You can use the new label to label files that you check in. This command does not attach the new label to any items unless you create the label by copying an existing revision label that is attached to one or more items. See the -vl option below.

-vd	Specifies the as-of date/time for the label when you create a view label. -vd is ignored if you are creating a revision label.
	Without the -vn or -vd or -vl option, the current time is used for view labels.
	Examples include:
	`"12/29/01 12:41 PM"`
	`"December 29, 2001 12:41:21 PM"`
	`"Monday, December 29, 2001 12:41"`

-vl	Specifies the name of the label to be copied. The name of the label is enclosed in double quotation marks. The specified label must already exist in the application and must be the same type as the label you are creating. If the specified label is a revision label that is attached to one or more items, the new label will be attached to those same items. Without the -vn or -vd or -vl option, the current time is used for view labels. If this is a revision label, make sure that you have set -r.
-vp	Specifies a promotion state whose label will be copied. The name of the state is enclosed in double quotation marks. The specified label must already exist in the application, and you must be creating a view label for this option to be valid. Without the -vn or -vd or -vl option, the current time is used for view labels.

Example

The following example uses stcmd label to create a new build label named Beta for the StarDraw view of the StarDraw project.

```
stcmd label -p "JMarsh:password@Orion:49201/StarDraw/StarDraw" -nl "Beta" -b
```

Locking and Unlocking Files: stcmd lck

Use stcmd lck to lock or unlock files in a view from the command line.

Syntax

stcmd lck	**-p** "*projectSpecifier*" [**-pwdfile** "*filePath*"] [**-cmp**] [**-csf**] [**-encrypt** *encryptionType*] [**-is**] [**-q]** [**-x]** [**-stop**] [**-rp** "*folderPath*" | **-fp** "*folderPath*"] [**-filter** "*fileStatus*"] [**-break**] [**-l** | **-u** | **-nel** [**-ro** | **-rw**]] [*files...*]
-break	Breaks the current lock by another user if you have the access rights to break locks.
-filter	Specifies a string of one or more characters, each of which represents a file status. Never include spaces or other whitespace in this string. Only files that currently have the specified statuses will be locked or unlocked. You cannot lock or unlock files that are Not In View. The letters used to represent the statuses are: C for Current, M for Modified, G for Merge, O for Out of Date, I for Missing, and U for Unknown.
-l	Locks the files. This is the default when -l, -nel or -u is not used.
-nel	Non-exclusively locks the files.
-ro	Makes the working file read-only after this operation. Without this option, the file remains as it was prior to the operation. Usually, you use -ro to prevent yourself from editing a file that is not locked by you. -ro must be used with -l or -u or -nel. If you use -ro, you cannot use -rw.
-rw	Makes the working file read-write after this operation. Without this option, the file remains as it was prior to the operation. -rw must be used with -l or -u or -nel. If you use -rw, you cannot use -ro.
-u	Unlocks the files.

Example

The following example uses stcmd lck to unlock all the files in SourceCode, a child of the root folder StarDraw (in the StarDraw view of the StarDraw project), as well as all the files in child folders of SourceCode.

```
stcmd lck -p "JMarsh:password@Orion:49201/StarDraw/StarDraw/SourceCode" -is -u
"*"
```

Listing Files: stcmd list

Use stcmd list to list all the files in the folder specified by the -p option. The files are those that existed at a specific time or that have a specific label.

Syntax

sstcmd list	**-p** "*projectSpecifier*" [**-pwdfile** "*filePath*"] [**-cmp**] [**-csf**] [**-encrypt** *encryptionType*] [**-is**] [**-q**] [**-x**] [**-stop**] [**-short**] [**-rp** "*folderPath*" | **-fp** "*folderPath*"] [**-cfgl** "*labelName*" | **-cfgp** "*stateName*" | **-cfgd** "*asOfDate*"] [**-filter** "*fileStatus*"] [**-cf**] [*files...*]
-cf	Causes the names of the child folders within the folder to be added to the list.
-cfgd	Configures the view as of the specified date/time. Examples include: "12/29/01 10:52 AM" "December 29, 2001 10:52:00 AM PST" "Monday, December 29, 2001 10:52:00 oclock AM PST"
-cfgl	Configures the view using the specified label. Without -cfgl or -cfgp or -cfgd, the view's current configuration is used.
-cfgp	Configures the view using the specified promotion state.
-filter	Specifies a string of one or more characters, each of which represents a file status. Never include spaces or other whitespace in this string. Only files that currently have the specified statuses will be listed. You cannot list files that are Not In View. The letters used to represent the statuses are: C for Current, M for Modified, G for Merge, O for Out of Date, I for Missing, and U for Unknown.

-short	Provides a short and simple listing of local files and their statuses consisting of the abbreviation for the status and the relative path to the working file, for example:
	`M /starteam/Server.java`
	`N /starteam/LabelInfo.java`
	Without this option, the listing consists of:
	A line for each folder name followed by its working folder's path
	Within a folder, a line for each file starting with the unabbreviated status and containing the rights, time stamp, and name of the file
	For example:
	`Folder: Source  (working dir: E:\Source)`
	`Unknown rw 4/6/02 7:42:18 PM PST  230 req.bmp`

Example

The following example uses stcmd list to list all the files in SourceCode, a child of the root folder StarDraw (in the StarDraw view of the StarDraw project), as well as all the files in child folders of SourceCode.

```
stcmd list -p "JMarsh:password@Orion:49201/StarDraw/StarDraw/SourceCode" -is "*"
```

Creating Working Folder: stcmd local-mkdir

Use stcmd local-mkdir to create the working folder or working directory on your workstation for the specified folder. Use -is to create the working folders (or working directories) for the specified folder's child folders as well.

Syntax

stcmd local-mkdir	**-p** "*projectSpecifier*" [**-pwdfile** "*filePath*"] [**-cmp**] [**-csf**] [**-encrypt** *encryptionType*] [**-is**] [**-q**] [**-x**] [**-stop**] [**-rp** "*folderPath*" |**-fp** "*folderPath*"] [**-cfgl** "*labelName*"	**-cfgp** "*stateName*"	**-cfgd** "*asOfDate*"]
-cfgd	Configures the view as of the specified date/time. Examples include: "12/29/01 10:52 AM" "December 29, 2001 10:52:00 AM PST" "Monday, December 29, 2001 10:52:00 oclock AM PST"
-cfgl	Configures the view using the specified label. Without -cfgl or -cfgp or -cfgd, the view's current configuration is used.
-cfgd	Configures the view as of the specified date/time. Examples include: "12/29/01 10:52 AM" "December 29, 2001 10:52:00 AM PST" "Monday, December 29, 2001 10:52:00 oclock AM PST"

Example

The following example uses stcmd local-mkdir to create the working folders for SourceCode, a child of the root folder StarDraw (in the StarDraw view of the StarDraw project) and its child folders.

```
stcmd local-mkdir -p "JMarsh:password@Orion:49201/StarDraw/StarDraw/SourceCode"
-is
```

Merging Files

Files can be merged as part of stcmd co, the check out command. See "Checking Out Files: stcmd co" on page 301.

You can also merge files using the View Comparison utility and the viewmerge command line. See the *StarTeam Administrator's Guide*.

Moving Status: stcmd move-status

The stcmd move-status can assist you as you transition from storing file status on the server in StarTeam 5.1 and earlier releases to storing file status on the client in StarTeam 5.2 and later releases.

The stcmd move-status command moves file status information from the server to the client. This command must be used only *before* you upgrade the Server to a post-5.2 release, and *after* you have upgraded the client to a post-5.2 release. After you upgrade the server, the status information will no longer reside on your server and there is no longer anything to move.

Before using the command, you must set your personal options to indicate where to store the client-side status information. It can be stored either on a per-folder or a centralized basis. Personal options can be set from the Windows or Cross-Platform client, or the command-line using stcmd set-personal-options. If no personal options have been set, all commands look for status information in the following centralized locations:

- For Windows NT, the folder in which the application has been installed.
- For Windows 2000 or XP:
  C:\Documents and Settings*username*\Local Settings\ Application Data\Borland\ StarTeam\syncdb
- For non-Windows platforms: */user_home_directory/*.starteam-client/syncdb

We recommend that you move the status information before upgrading the server. Otherwise, all the file statuses become Unknown.

Many statuses can be recalculated afterwards using the stcmd update-status command at the command line or the File\Update Status command in the Windows or Cross-Platform client, but the Merge status, one that is of considerable importance, cannot be recalculated. Also, the recalculation takes longer than the move-status operation.

As an alternative, you can make sure that all files are current before the server is upgraded to a post-5.2 release.

Syntax

stcmd move-status	**-p** "*projectSpecifier*" [**-pwdfile** "*filePath*"] [**-cmp**] [**-csf**] [**-encrypt** *encryptionType*] [**-is**] [**-q**] [**-x**] [**-stop**] [**-rp** "*folderPath*" | **-fp** "*folderPath*"] [**-cfgl** "*labelName*" | **-cfgp** "*stateName*" | **-cfgd** "*asOfDate*"] [**-filter** "*fileStatus*"] [*files...*]
-cfgd	Configures the view as of the specified date/time. Examples include: "12/29/01 10:52 AM" "December 29, 2001 10:52:00 AM PST" "Monday, December 29, 2001 10:52:00 oclock AM PST"
-cfgl	Configures the view using the specified label. Without -cfgl or -cfgp or -cfgd, the view's current configuration is used.
-cfgp	Configures the view using the specified promotion state.
-filter	Specifies a string of one or more characters, each of which represents a file status to be moved. Never include spaces or other whitespace in this string. Only files that currently have the specified statuses will have their statuses moved to the client. The Not In View status is not in the list because the application does not store information about files with this status. The letters used to represent the statuses are: C for Current, M for Modified, G for Merge, O for Out of Date, I for Missing, and U for Unknown.

Example

The following example uses stcmd move-status to move all status information for the files in the SourceCode folder and its child folders from the server to the client.

```
stcmd move-status -p "JMarsh:password@Orion:49201/StarDraw/StarDraw/
SourceCode" -is "*"
```

Removing Files: stcmd remove

Use stcmd remove to remove files from version control. The specified files and their revision histories no longer appear in the application unless you roll back the project view to a time before they were removed.

Syntax

stcmd remove	**-p** "*projectSpecifier*" [**-pwdfile** "*filePath*"] [**-cmp**] [**-csf**] [**-encrypt** *encryptionType*] [**-is**] [**-q**] [**-x**] [**-stop**] [**-rp** "*folderPath*" | **-fp** "*folderPath*"] [**-filter** "*fileStatus*"] [**-df**] [*files...*]
-df	Deletes the user's working file. Without this option the working file remains in the working folder on your workstation.
-filter	Specifies a string of one or more characters, each of which represents a file status. Never include spaces or other whitespace in this string. Only files that currently have the specified statuses will be removed. You cannot remove files that are Not In View from version control. The letters used to represent the statuses are: C for Current, M for Modified, G for Merge, O for Out of Date, I for Missing, and U for Unknown.

Example

The following example uses stcmd remove to remove all the .hm files from SourceCode, a child of the root folder StarDraw (in the StarDraw view of the StarDraw project), as well as all the files in child folders of SourceCode. It also deletes the working files.

```
stcmd remove -p "JMarsh:password@Orion:49201/StarDraw/StarDraw/SourceCode" -is
-df "*.hm"
```

Locking/Unlocking a Server: stcmd server-mode

Use stcmd server-mode to lock and unlock a server—if you have the appropriate access rights or privileges. Locking a server limits access to it while you perform backup or other procedures. When the server is locked, only server administration commands are accepted. When you unlock the server, normal operations resume.

Note Do not use the user name “StarTeam” with this command, as it requests a password for the user even when one has already been provided or when the user has a blank password.

Syntax

stcmd server-mode	[**-pwdfile** "*filePath*"] [**-cmp**] [**-encrypt** *encryptionType*] [**-q**] [**-x**] [**-stop] -s** "*serverName*" **-mode** [**lock | exlock | unlock**]
exlock	Exclusively locks the server so that no one else can access it.
lock	Nonexclusively locks the server. Only administrative commands can be performed.
-mode	Indicates whether the server is to be locked, exclusively locked, or unlocked. If you use -mode lock, only server administration commands are accepted until the server is unlocked. For example, you might use this command while running a backup program. If you use -mode exlock, only you can access the server until it is unlocked. For example, you might do this when creating a custom field. Use -mode unlock to make the server available to users again.
unlock	Unlocks the server so that anyone with the appropriate access rights can access it.
-s	Specifies the server. The full syntax is: **-s** "*userName*:*password*@*hostName*:*portNumber*" For example: -s "JMarsh:password@orion:49201" If the user name is omitted, the current user name is used. The user name in the example is “JMarsh” If the password is omitted, the user is prompted to enter the password. The password in the example is “password” If the host name is omitted, the default is localhost. The host name in the example is “orion”. The port number is required. The default port number, 49201, is used in the example.

Example

The following example uses stcmd server-mode to nonexclusively lock the Orion server on port 49201:

```
stcmd server-mode -s "JMarsh:password@Orion:49201" -mode lock
```

Setting Personal Options: stcmd set-personal-options

Use stcmd set-personal-options to set and list personal options. At this point in time, the only personal options that can be set are:

- the way in which file status information will be stored
- the location for storing the information when it is stored at a central location

This information is stored in the starteam-client-options.xml file. If no starteam-client-options.xml file exists, this command will create it in a default location. The default location is the same as the default for the central-status option (listed below).

Syntax

stcmd set-personal-options	[**-q**] [**-x**] [**-stop**] [**-central-status** | **-per-folder-status**] [**-central-repository** "*folderPath*"] [**-list**]
-central-repository	Enables you to specify a location for the central repository.
-central-status	Indicates that the file status information will be stored at a central location for this user on this workstation. This location can be set using the Windows or Cross-Platform client, or this command (see the -central-repository option). If you do not set this location it defaults to: ■ For Windows NT, the folder in which the application has been installed. ■ For Windows 2000 or XP: C:\Documents and Settings*username*\Local Settings\Application Data\Borland\StarTeam\ ■ For non-Windows platforms: */user_home_directory/*.starteam-client/
-per-folder-status	Indicates that each working folder will contain file status information for its own files. The information is stored in the .sbas folder, a child folder of the working folder. If you move the working folder, the file status information goes with it.
-list	Displays a list of the personal options as currently set in the starteam-client-options.xml that stores them.

Example

The following example uses stcmd set-personal-options to indicate that file status information will be stored in a central location: C:\jmarsh\statusinfo.

```
stcmd set-personal-options -central-status
-central-repository "C:\jmarsh\statusinfo"
```

Updating File Status: stcmd update-status

When you update the status of a file, the application compares the working file with the revision you checked out and the tip revision. For example, the application may say that a file is Current, but someone else has just checked in a copy of that file, so your status is really Out Of Date.

Updating file statuses is not the same as updating files. For example, if a file is not in your working folder, updating the status will let you know that the file's status is Missing. It will not check out the file for you so that it is no longer missing. After all, you may not want the file on your hard drive. Normally, you use a file's status to determine whether the file should be checked in, checked out, added, or ignored.

For example, you may want to:

- Check in a file if its status is Out Of Date, Missing, or Merge.
- Check out a file if its status is Modified or Merge.
- Add a file to the application if its status is Not In View. However, the update-status command never lists files that have the status Not In View because they are not stored in the repository.

For a complete list of file statuses, see "Determining the Status of Your Files" on page 130.

Use stcmd update-status to display the file name, its status before the command, and its status after the command. For example, one line of output might be:

```
x.cpp: status is Current (was Unknown)
```

Syntax

stcmd update-status	**-p** "*projectSpecifier*" [**-pwdfile** "*filePath*"] [**-cmp**] [**-csf**] [**-encrypt** *encryptionType*] [**-is**] [**-q**] [**-x**] [**-stop**] [**-rp** "*folderPath*" |**-fp** "*folderPath*"] [**-cfgl** "*labelName*"| **-cfgp** "*stateName*" | **-cfgd** "*asOfDate*"] [**-filter** "*fileStatus*"] [**-eol** [**on** | **off**]] [**-v**] [*files...*]
-cfgd	Configures the view as of the specified date/time. Examples include: "12/29/01 10:52 AM" "December 29, 2001 10:52:00 AM PST" "Monday, December 29, 2001 10:52:00 oclock AM PST"
-cfgl	Configures the view using the specified label. Without -cfgl or -cfgp or -cfgd, the view's current configuration is used.
-cfgp	Configures the view using the specified promotion state.
-eol	When on, calculates status of text file without using end-of-line (eol) markers. When off, which is the default, the update status is computed on working file and tip revisions as the are with their current end-of-line markers. With this option on, a working file that has LF eol markers can be compared to a tip revision with CR/LF eol markers and be considered Current if the only difference is the eol markers.
-filter	Specifies a string of one or more characters, each of which represents a file status. Never include spaces or other whitespace in this string. Only files that currently have the specified statuses will be updated. C for Current, M for Modified, G for Merge, O for Out of Date, I for Missing, and U for Unknown.
-v	Reports the status of every file in the specified folder's working folder—unless its status is Not In View. Without this option, a file's status is displayed only if it has changed.

Example

The following example uses stcmd update-status to make sure that each file in the working folder for the folder named SourceCode has an accurate status. SourceCode is a child of the root folder StarDraw (in the StarDraw view of the StarDraw project).

```
stcmd update-status -p "JMarsh:password@Orion:49201/StarDraw/StarDraw/
SourceCode" "*"
```

Chapter

23

Using the bco Command Line Interface

With the Bulk Check-out (bco) utility, which operates with the Windows and Cross-Platform client, you can quickly check out a large number of files from an application repository to your working folders. You may want to use this utility to check out files for builds, as it is faster than the stcmd co (page 301), the standard application command-line check-out, especially when combined with Cache Agent.

Most build processes check-out from a view label or promotion state (using -cfgl and -cfgp). If you check-out tip revisions, be aware that users may be checking in the files that you are checking out, causing an inconsistent snapshot. You must use other means to ensure a consistent snapshot, if you need one. For example, you can exclusively lock the server using the stcmd server-mode command or ask users not to check in files until further notice.

The Bulk Check-out utility does not recognize differences in folder names due to case-sensitivity. It does recognize these differences in file names. For most users, this should not be a problem.

Syntax

bco

```
-p "projectSpecifier" [-pwdfile "filePath"] [-autoLogon] [-is]
[-rp "folderPath" | -fp "folderPath" ]
[-cmp]  [-dryrun] [-vb] [-useCA]
[-encrypt encryptionType]
[-cfgl "labelName"| -cfgp "stateName"
| -cfgd "asOfDate"]
[-filter "fileStatus"]
[-o] [-ro] [-ts] [-fs]
[-eol on | off | cr| lf| crlf ]
[-netmon] [-t] [-h | -help]
[files...]
```

-autoLogon

If a user name is not specified in the -p option, an attempt is made to logon using the user ID and password for the specified Server as stored by the Toolbar utility. This is available only on Windows operating systems.

-cfgd	Configures the view as of the specified date/time. Examples include: ■ "12/29/01 10:52 AM" ■ "December 29, 2001 10:52:00 AM PST" ■ "Monday, December 29, 2001 10:52:00 oclock AM PST"
-cfgl	Configures the view using the specified label. Without -cfgl or -cfgp or -cfgd, the view's current configuration is used.
-cfgp	Configures the view using the specified promotion state.
-cmp	Compresses all the data sent between the workstation and the server and decompresses it when it arrives. Without this option, no compression takes place. Compression is most useful and appropriate when the client and server communicate over a slow connection. To determine whether to use compression, a small test case may be helpful. You must consider whether the time spent compressing and uncompressing data is better than the longer transfer time of uncompressed data sent over the slow connection.
-dryrun	Does not check out files, but displays a list of the files that would be checked out if -dryrun were not specified. The paths are those for the working folders into which the files would have been copied. When used with -vb, you get a complete picture of what would have happened.
-encrypt	Encrypts all the data sent between the workstation and the server and unencrypts it when it arrives. Without this option, no encryption takes place. Encryption protects files and other project information from being read by unauthorized parties over unsecured network lines. The full syntax is: **`-encrypt`** *`encryptionType`* The types of encryption are: **RC4** RSA R4 stream cipher (fast) **RC2_ECB**RSA R2 block cipher (Electronic Codebook) **RC2_CBC**RSA R2 block cipher (Cipher Block Chaining) **RC2_CFB** (Windows platforms only) RSA R2 block cipher (Cipher Feedback) These encryption types are ordered from fastest to slowest. Each of the slower encryption types is safer than the one preceding it.
-eol	Can automatically convert end-of-line markers. When on, text files are transferred from the Server's repository to the workstation's working folder with the end-of-line convention for the platform executing the command as determined by the Java VM. When off, the default, no end-of-line conversion is performed. Using off is the same as not using -eol at all. When you specify the end-of-line character (cr, lf, or crlf), text files are transferred from the Server's repository to the workstation's working folder with the specified end-of-line convention. For Windows clients, the end-of-line marker is a carriage return/line feed (crlf) combination; for UNIX platforms, it is a line feed (lf); for MAC systems, a carriage return (cr). You would set this option to on or lf, for example, when you compare a file from the repository and a working file on a UNIX system (if the repository stores text files as crlf).

files...

Specifies the files to be used in the command by name or by file name-pattern specification (such as "*.c"). All options are interpreted using the semantic conventions of UNIX instead of Windows because UNIX's conventions are more specific. This means that "*", rather than "*.*" means "all files." The pattern "*.*" means "all files with file name extensions." For example, "star*.*" finds starteam.doc and starteam.cpp but not starteam. To find all of these, you could use "star*".

Without this option, the default is "*". When used, this option must always be the last option. Any options after it are ignored.

If you use * rather than "*" to indicate all files, a UNIX shell expands it into a series of items and passes this series as a group of options to the bco command. This can cause problems, for example, when you are checking out missing files, so it is best to use "*" to avoid unwanted complications.

If you use a set of file patterns, each pattern should be enclosed in its own set of quotation marks. For example, you can use "*.bat" "*.c", but you cannot use "*.bat *.c".

- **Note:** We recommend that you enclose this option in quotation marks, regardless of platform, but for different reasons. On Windows platforms, file and folder names that contain spaces will not be interpreted correctly. On UNIX platforms, the shell will expand the option, then pass the list of items produced by the expansion to the client. Frequently, this produces unintended results. You can avoid both of these consequences by always enclosing this option in quotation marks. Only if it is essential that the option be expanded by the UNIX shell is it advisable to omit the quotation marks. Mysterious failures of batch scripts on either platform may be due to this omission.

Several special characters can be used in the file specification:

*****

Matches any string including the empty string. For example, "*" matches any file name, with or without an extension. "xyz*" will match "xyz" and "xyz.cpp" and "xyzutyfj".

?

Matches any single character. For example, "a?c" will match "abc" but NOT "ac"

[...]

Matches any one of the characters enclosed by the left and right brackets.

A pair of characters separated by a hyphen (-) specifies a range of characters to be matched.
If the first character following the right bracket ([ ) is an exclamation point (!) or a caret (^), the rest of the characters are not matched. Any character not enclosed in the brackets is matched. For example, "x[a-d]y" matches "xby" but not "xey". "x[!a-d]y" matches "xey" but not "xby".

A hyphen (-) or right bracket ( ]) may be matched by including it as the first or last character in the bracketed set.

To use an asterisk (*), question mark (?), or left bracket ([ ) in a pattern you must precede it with the escape character, which is the backslash \ .

-filter

Specifies a string of one or more characters, each of which represents a file status. Never include spaces or other whitespace in this string. Only files that currently have the specified statuses will be checked out. You cannot check out files that are Not In View.

The letters used to represent the statuses are:
C for Current, M for Modified, G for Merge, O for Out of Date, I for Missing, and U for Unknown.

If you use G, M, O, or U, you must also specify -o to force the check-out operation. Without -o, files with the statuses represented by G, M, O and U are skipped, but a warning is logged in the log file.

-fp

Overrides the specified folder's working folder or working directory. This is equivalent to setting an alternate working path for the folder.

While this enables you to use a different working folder than the one specified by the folder, *its critical importance is its use to provide cross-platform compatibility*. For example, UNIX and Windows systems specify drive and directory path names in incompatible ways. Although the path "D:\MYPRODUCT\DEVELOPMENT\SOURCE" is understood on a Windows platform, it is *not* understood on a UNIX platform. Use this option to define the working path if your platform does not understand the path specified in the project.

A backslash (\) is interpreted as an escape character when it precedes quotation marks. As a result, an error occurs in the following example:

```
bco -p "xxx" -fp "C:\" "*"
```

which is interpreted as:

```
bco -p "xxx" -fp "C:" *"
```

To avoid a situation like this, escape the final character in `"C:\"` as follows:

```
bco -p "xxx" -fp "C:\\" "*"
```

Or avoid it as follows when the -fp path does not end with the root folder as in "C:\orion\":

```
bco -p "xxx" -fp "C:\orion" "*"
```

The full syntax is:

`-fp` ***`"folderPath"`***

Folder is the Windows term and appears in the user interface. Directory is the correct term for the UNIX platform.

-fs

Prevents file statuses from being remembered after the check-out occurs. Subsequent status values for these files will be incorrect and indeterminate. Use this option where a file's status is irrelevant. For example, if you routinely delete the working folders before checking out files for a build, there are no files and their statuses do not matter.

Be aware that the file statuses may never be known—even if you use the update-status command later. You can do a force check out without the -fs option to obtain current files with correct statuses.

-h
-help

Displays information about the options.

-is

Applies the command to recursively to all child folders. Without this option, the command applies only to the specified folder.

-netmon

Outputs SDK NetMonitor information to the console window. NetMonitor displays statistics for server commands. See the examples for bco at the end of this chapter.

-o

Forces the check-out of files whose status would normally not allow them to be checked out. Those statuses are Modified, Merge, or Unknown.

-p

Indicates what view or folder is to be used, as well a providing the user name and password needed to access the server.

The full syntax is:

`-p "`*userName*`:`*password*`@`*hostName*`:`*portNumber*`/`*projectName*`/[`*viewName*`/][`*folderHierarchy*`/]"`

For example:

-p "JMarsh:password@orion:49201/StarDraw/StarDraw/SourceCode/"

If the user name is omitted, the current user name is used. The user name in the example is “JMarsh”

If the password is omitted, the user is prompted to enter the password. When the user types a password, the characters are not displayed on the screen. The password in the example is “password”.

If the host name is omitted, the default is localhost. The host name in the example is “orion”.

The port number is required. The default port number, 49201, is used in the example.

The project name is always required. The project name in the example is StarDraw.

Use a view hierarchy to identify the view. Use the colon (:) as a delimiter between view names. The view hierarchy should *always* include the root view. For example, "StarDraw:Release 4:Service Packs" indicates that the view to be used is the Service Packs view, which is a child of the Release 4 view and a grandchild of the StarDraw root view. If the view name is omitted, the root (default) view is used. If the view is the only view in that project with that name, you can use only the view name. (This is not recommended because another view with that name could be created at a later date and cause confusion.) The view name in the example is StarDraw. Because this is the root view of the StarDraw project, it could have been omitted.

Use a folder hierarchy to identify the folder. Use the forward slash (/) as a delimiter between folder names. The folder hierarchy *never* includes the root folder. Omit the folder hierarchy if the file is in the view’s root folder. For example, if the root folder of the view is StarDraw and the hierarchy to your files is "StarDraw/SourceCode/Client", use only "SourceCode/Client".

If any of the variables used with this option, for example, user names or view names, contain characters that are used as delimiters, please use the percent sign (%) followed by the hex code for each of those characters. For example, if "@" appears as a character in a password, you must replace it with "%40".

For ":", use "%3a".

For "/", use "%2f".

For "@", use "%40".

For "%", use "%25

In UNIX and other operating systems, some special characters must be preceded by a backslash "\" or another escape character. In the -p option, you can replace such characters with hex codes. For example, "%3c" could be used in UNIX instead of "\<".

For a space, use "%20".

For "<", use "%3c".

For ">", use "%3e".

-pwdfile	Specifies the path to a file that stores the user's password. This option overrides the password used as part of the -p option. It prevents others from seeing the user's password on the command line. It must be saved in UTF-8 format. The full syntax is: -pwdfile "*filePath*" The password file should consist only of the password. Leading and trailing whitespace is ignored.
-ro	Makes the working file read-only after this operation. Without this option, the file remains as it was prior to the operation. Usually, you use -ro to prevent yourself from editing a file that is not locked by you. Without -ro, the files are read/write.
-rp	Specifies or overrides the working folder or working directory for the view's root folder. This is equivalent to setting an alternate working path for the view. See "-fp" on page 322 for additional information. The full syntax is: **-rp "***folderPath***"** Specifies or overrides the working folder or working directory for the view's root folder. This is equivalent to setting an alternate working path for the view. See "-fp" on page 322 for additional information. The full syntax is: -rp **"***folderPath***"**
-t	Displays check-out volume and timing statistics.
-ts	Sets each working file's time stamp to the check-out time. Without this option, the file is given the same time stamp as the checked-in revision of the file.
-useCA	Attempts to check-out files using a StarTeamMPX Cache Agent. The full syntax is: -useCA *hostName*:*portNumber* | autolocate If you specify the host name (or IP address) and port number of the Cache Agent to be used, you do not have to have the StarTeam Server MPX-enabled at the time you run the BCO utility. If you use autolocate instead of specifying the host name and port number, the utility can automatically locate the network-nearest Cache Agent, but only if the server is MPX-enabled.
-vb	Output is verbose. Displays each file as it is checked-out. The folder path is the folder path, rather than the working folder path.

Example

The following example uses bco to force check out all the files from Source Code, a child of the root folder StarDraw (in the StarDraw view of the StarDraw project).

```
bco -p "JMarsh:password@Orion:1024/StarDraw/StarDraw/Source Code" -is -o "*"
```

The next example shows a bco command that uses -netmon and the output displayed by Net Monitor.

```
bco -p "Administrator:Administrator@10.50.6.91:49201/StarDraw/WebSite" -fp D:\
Test -netmon -o "*.htm"
```

Output:

```
StarTeam BulkCheckOut Utility version 7.0.xxx
Copyright (c) 2004 Borland Software Corporation. All rights reserved.
Start: (rev 100) SRVR_CMD_GET_PROJECT_LIST Time: 62 millis; Sent: 42 bytes;
Got: 1834 bytes
Start: (rev 100) SRVR_CMD_GET_PROJECT_VIEWS Time: 47 millis; Sent: 46 bytes;
Got: 186 bytes
```

```
Start: (rev 100) SRVR_CMD_GET_PROJECT_VIEWS Time: 15 millis; Sent: 46 bytes;
Got: 186 bytes
Start: (rev 100) SRVR_CMD_PROJECT_OPEN Time: 188 millis; Sent: 70 bytes; Got:
120 bytes
Start: (rev 100) PROJ_CMD_GET_VIEW_PROPERTIES Time: 31 millis; Sent: 42 bytes;
Got: 2556 bytes
Start: (rev 100) PROJ_CMD_GET_FOLDERS Time: 63 millis; Sent: 42 bytes; Got:
1112 bytes
Start: (rev 100) PROJ_CMD_GET_FOLDER_ITEMS Time: 16 millis; Sent: 50 bytes;
Got: 40 bytes
Start: (rev 100) PROJ_CMD_REFRESH_ITEMS Time: 3562 millis; Sent: 122 bytes;
Got: 414 bytes
Start: (rev 100) SRVR_CMD_GET_PROJECT_VIEWS Time: 16 millis; Sent: 46 bytes;
Got: 186 bytes
Start: (rev 100) PROJ_CMD_GET_PROJECT_PROPERTIES Time: 31 millis; Sent: 42
bytes; Got: 4797 bytes
Start: (rev 100) FILE_CMD_CHECKOUT Time: 47 millis; Sent: 78 bytes; Got: 108
bytes
Start: (rev 100) FILE_CMD_CHECKOUT Time: 31 millis; Sent: 78 bytes; Got: 1767
bytes
Start: (rev 100) FILE_CMD_CHECKOUT Time: 31 millis; Sent: 78 bytes; Got: 1140
bytes
Start: (rev 100) SRVR_CMD_PROJECT_CLOSE Time: 15 millis; Sent: 62 bytes; Got:
16 bytes
Start: (rev 100) SRVR_CMD_RELEASE_CLIENT Time: 31 millis; Sent: 42 bytes; Got:
16 bytes
```

Appendix

A

Glossary

access rights	A security feature. The rights granted (or denied in exceptional cases) to users or groups that allow team members to see items, modify them, create them, and so on.
add files	The process of placing files under version control by adding them to a project view.
administrator	A functional role in the operation of the application. Administrators create and/or import projects, manage server configurations, users and groups, and so on.
All Descendants	The button at the top of the upper pane. Also a command on the File, Change Request, Task, Topic, and Audit menus. When selected, the view window displays information for the selected folder and its child folders. Otherwise, the view window displays only the items associated with the folder and not with its child folders.
alphanumeric	A value consisting of the letters A-Z, and the digits 0-9.
alternate working folder	Creating an alternate working folder enables you to store that folder's working files on your workstation at the location you specify. Creating an alternate working folder for the root of a view or a branch in a folder hierarchy can alter the paths of the working folders for child folders.
archive	The file or group of files that make it possible to recreate past revisions of a file that is under version control.
ASCII file	A text file. The application recognizes a text file by its lack of null characters.
audit entry	A record of an action performed on a project that appears in the audit log. For example, every time a file is added to a project view, that action is entered in the audit log.
audit log	A chronological record kept by the application showing all actions performed on folders and items.
author	The person who created the revision.
automatic refresh	An automatic update feature of the application. Because projects, folders, and items managed by the application are subject to continuous revision by various team members, screen information can become outdated after it is first displayed. The Automatic Refresh function periodically reads the project database to update the displayed information.

binary file	Any file that is not strictly an ASCII text file. Binary files contain programs or data in machine code. Examples of binary files and their respective file name extensions are executables (.exe and .dll), word processing documents (.doc in Word for Windows), spreadsheets (.xls in Excel), object files (.obj) and bitmaps (.bmp). The application identifies text files as any files that contain no null characters. All other files are binary.
blank branching view	An empty branching view. It has no correlation to the parent view.
branch	The process of creating an independent item that is derived from a corresponding item in a parent view. In the case of a text file, the branched item can later be merged with the file from which it originated. For example, the development of a product for a new operating system may start with the existing files for the first operating system as its base. See also branching view. Also a branch of a tree, such as the folder hierarchy or a topic tree.
branch revision	The revision number is an identification number assigned by the application to a revision of an item. It indicates how many consecutive revisions have been made in a view and branches as the item is branched into a new view. For example, if a branching view is created that includes a change request with the revision number 1.9, the next revision of that change request in the new view will probably be 1.9.1.0.
branching view	A view that may or may not be derived from a parent view. When not derived from a parent view, it is a blank branching view. Branching views always permit branching. If they float and have the Branch On Change option set, they are updated by the parent view on a file-by-file basis until that file changes in the branching view. If they float and do not have the Branch On Change option set, updates are sent to the parent view until a pint in time when the Branch On Change option becomes set. If they are based on a label, a promotion state, or a moment in time, they are read-only unless the Branch On Change option is set.
build	The process of compiling, assembling, and linking all project files in proper sequence to produce the project´s Also an event in the life cycle of a product chosen to represent a quantifiable step in progress for a project. For example, a software product may deliver a new build every few days to the team members who test the product. Applying a build label to a set of revisions at the view level ensures that every item in that view at the time of the build has the same label. For example, this makes it possible to identify what revisions of source code files product were n each build.
build label	When you create a view label, you can designate it as a build label. Build labels change the setting for the Addressed In field for newly resolved change requests from Next Build to the build label. This lets testers know that a resolved change request can be tested by installing that build.
change request list	The list of change requests, related to your selection from the folder hierarchy, that is displayed when you select the Change Request tab. The list is further refined by the All Descendants button and filter you select.
change request number	The number that the application assigns to a change request when it is first submitted.
check-in	The operation that stores a new file revision as an archive file in the application and data about the file in the repository. The application permits a number of individuals to work on a common set of files by allowing only one team member to revise a project file at a time. Check-in marks the end of one revision. The team member who checks in the file can keep it locked or release the file to others by unlocking it.
check-out	The operation that copies a revision of a file from the application archive to a team member's working folder. A team member can check out a file with or without the intention to alter that file. The application permits a number of individuals to work on a common set of files by allowing only one team member to revise a project file at a time. Locking the file marks the beginning of one author's revision.

column header	The label or name for a column that appear in the upper pane of the project view window. Clicking the header initiates sort operations based on the values in that column. Each column header is the name of a property for the items in the list.
command-line file	When you compare two files or revisions, The application sends a command-line file to Visual Diff containing the appropriate options. A similar file is sent to Visual Merge when you merge two files. If there is no space for this file or no temp folder to create it in, an error message appears.
component	This manual refers to parts of the application as components. For example, it references the File component or Change Request component. The files, change requests, etc. managed by the component are referred to as items.
compression	Data that is transferred between your workstation and the server can be compressed. Data compression reduces the amount of traffic on the network. However, the time to compress and decompress the data is added to the transfer time.
configuration	A relative arrangement of parts or elements. The application has view, folder, item and server configurations. A view, folder, or item configuration is the isolation of that view, folder, or item to a particular revision based on a point in time. For example, you can configure a view to: ■ Be current (so that you always see the tip revisions of every folder and item in the view). ■ A view label (so that you see all the revisions in the view that have the selected label assigned to them). A view label initially represents a point in time although the label can be adjusted to include revisions that were not current at that point in time and exclude revisions that were. ■ A promotion state (so that you can see all the revisions in the view that have been assigned the label that is currently associated with the selected promotion state). ■ Any selected date and time (so you call see all the revisions in the view that were current at the specified date and time). You can also configure individual folders and items. See also server configuration.
container	A term indicating the ability to contain other types of items. For example a project is a container for views, folders, and items. Views and folders are also containers.
context menu	A menu that appears when you right-click an area in the project view window. The menu contents vary with the context, that is, with which item or area has been clicked.
Current	File status. The content of the file in the working folder is the same as the content of the tip revision of this file.
defect	A fault or error in a product.
defect tracking	A systematic way of recording information about change requests and maintaining a history of their detection and eventual resolution.
delete	To remove information from application databases. You can delete: ■ Items within the project. They can become visible again by rolling back the view or parent folder to a time prior to the deletion. ■ Working files. When these are deleted, they are gone. If the working file was checked out and unchanged, added, and/or check-in before the deletion, no data was lost. Delete items with care.

delta storage	A method of computing differences between progressive revisions of a file. In StarTeam 6.0 and earlier, the application could store the first revision of a text file in its entirety in the Archive, along with a set of forward deltas for later revisions. This use of delta storage disappears with StarTeam 2005. StarTeam 6.0 and earlier releases also used deltas to optimize for slow connections if users selected that option. StarTeam Server 2005 and later releases continue this practice, but store these deltas in the Delta folder for each hive, which is a subfolder of each hive's cache path. See also *full revision storage*.
e-mail	An organized system for the delivery of paperless, “electronic” messages, named for its similarities to the postal mail system. The application provides advantages (such as mailing items) for teams that use a MAPI-compliant e-mail system.
encryption	Data that is transferred between your workstation and the server can be encrypted. Encryption protects files and other project information from being read by unauthorized parties over unsecured network lines.
endpoint	A unique identifier for a server, based on the protocol being used. The endpoint is a port number for the TCP/IP protocol.
exclusive lock	A lock, indicating that you are editing a specific file. For a file, the icon others see is yellow padlock, while you see an icon with a small yellow key and the head and shoulders of a person. This lock notifies others that you intend to revise this file. Others cannot check in a file that you have exclusively locked.
file compression	A technique for reducing the size of a file by removing redundant information from it. Most disk files contain repetitions of common sequences of characters. Compression algorithms remove the additional occurrences of these sequences and save information that permits their restoration.
filter	The criteria used to select a few items from among many. The Filter drop-down list box in the view window enables you to display only the items that are of interest. Applying a filter also controls what columns are displayed, what columns are sorted and how to group values in the sorted columns.
fixed change request	A designation for a change request indicating that it has been resolved.
float	A view is said to float if it stays current with its parent view. That means updates from the parent are sent to the child view. If the child is not read-only, updates from the child also go to the parent.
folder	See folder hierarchy and working folder.
folder hierarchy	The hierarchical display of a view and its associated folders. The folder hierarchy is always displayed in the left pane of the view window.
foreign refresh	Pressing F6 enables you to update the information in your view window for a PCVS or SourceSafe project with which you are communicating via the application.
frozen	An item is said to be frozen (and, therefore, read-only) if it is based on the state of its corresponding item in the parent view at a specific moment in time AND cannot be branched. An item is also frozen if you reconfigure it to a specific label, promotion state, or time in its past.
full revision storage	A storage method in which all revisions of a file are saved in their entirety. In StarTeam 6.0 and earlier, choosing full revision storage reduced the processing time necessary to reconstruct complex biary files stored in delta format in the Archive. In StarTeam 2005 and later releases, all archive files are stored in their entirety, although some may be compressed. See delta storage.
history list	The list of revisions for the item selected from the list in the upper pane of the view window. The application displays this list when you select the History tab from the view window’s lower pane.
item	An object or element. Items include projects, views, folders, files, change requests, requirements, tasks, topics, and audit entries.
keep current lock status	An option that does not allow the lock status (exclusive, non-exclusive, or unlocked) to be changed as part of the current operation.

keyword	Reserved words beginning and ending with a dollar sign ($). When used in a text file, the application replaces them with the data that they represent. For example $Author$ is replaced by the name of the user who checked in the file.
keyword expansion	A technique used to insert information in a text file in which keywords are replaced by the data they represent. For example $Author$ is replaced by the name of the user who checked in the files.
label	The act of attaching a view or revision label for one or more folders and/or items. See build label, revision label, and view label.
labeled configuration	The basis for a new view or a reconfigured view. The view contains only items with the label you specify. This option is disabled in the new view's parent view or the reconfigured view has no labels defined for it.
lock	File locking is a technique to inform others that you are revising a file (see exclusive lock) or thinking about revising it (non-exclusive lock). Checking files in and out does not imply locking in the application automatically. You can lock or unlock a file during check-in or check-out manually. However, you can also lock or unlock files with the Lock/Unlock dialog. Locks can be overridden and can be broken. Depending on your personal options, you may have unlocked files marked read-only, which prevents them from making any changes to files that may be locked by others.
MAPI	Acronym for Mail Application Programming Interface). A programming interface that permits an application to send and receive electronic mail via the Microsoft Mail messaging system. The application uses SMTP and not MAPI.
Merge	The file status that indicates that the working file is not based on the tip revision of the file. As you check this file in or out, the application asks if you want to merge it with the tip revision.
merge	The process of combining a working file with the tip revision of that file or of combining a branched file with the original file from which it was branched. This is a three-way comparison, in which both files are compared with the file revision that is their most recent common ancestor. The combined file can be inspected or revised by the user before it is checked in.
milestone	An event in the life cycle of a product chosen to represent a significant step in progress, for example, the alpha, beta, or final release of a product. In the application, when you reach a milestone, you can apply a view label, usually a build label, to indicate that the milestone has been reached. In Microsoft Project, a milestone is a type of task that represents a significant landmark, development, or turning point in the life of a scheduled project. You usually define a milestone by entering a task name and assigning it a duration of zero. Milestones typically signal that the work has started or is completed and do not represent the doing of the work.
Missing	File status. The file is not in your working folder. You might want to check the file out.
Modified	File status. The working file has been altered and is based on the tip revision of this file. You might want to check this file in.
non-exclusive lock	A lock indicating that you are thinking about changing the file. It notifies others with similar intentions to talk to you before they lock the file exclusively and change it. While you have a file locked non-exclusively, others can check the file in. For text files, it is not unusual for several people to modify the file simultaneously and rely on Visual Merge to manage the changes unless there are conflicts. In that case, a non-exclusive lock helps you identify whose input you need in order to resolve a conflict.
Not In View	File status. The file is in the working folder, but is not in the view. You might want to add this file to the view.

notification of events	If you use system tray notification, the application notifies you when an item needs to be read by displaying the appropriate icon in the system tray. If your server is set up for e-mail notification, you receive e-mail notification about items for which you are responsible and so on.
numeric	A value consisting of the valid digits 0-9.
object	A generic term used in object-oriented programming and elsewhere to indicate something upon which operations can be performed. See item.
Out Of Date	File status. The working file is a copy of an old revision of the file. It you need the current revision, you should check it out.
owner	Information stored by the application about the user who performed a certain initial operations on an item. ■ For a file: The user who added the file to the project view. ■ For a revision: The user who checked in the revision. ■ For a change request: The user who submitted the change request. ■ For a topic or response: The user who wrote the topic or response. ■ For a requirement or task: The user who created the task.
pane	The client's project view window is divided into three panes. The left pane displays the hierarchy of folders in the current view of the project. The upper pane lists items and the lower pane displays the detail, history or link information corresponding to your selection from the upper pane.
personal options	User-selectable choices for the behavior of the application for the currently logged-on user on a specific workstation.
profile	A set of limitations applied to a view. A profile specifies which folders are visible and which EOL and path case sensitivity settings apply to those folders. For example, if you are building a UNIX version and a Windows version of a product from the same view, you need a profile for each. Every view starts with the Default profile, but new profiles can be created with default settings or based on previous profiles. Any changes made to a folder's visibility, EOL, and path case sensitivity settings automatically become part of the currently selected profile.
project	A set of related views (each of which usually represents a version of a single product under application version control), project properties and project access rights.
promotional model	The application provides an object-oriented architecture which supports entity-relationship modeling. The application enables you to move (promote) changes between different stages of the project, for example from development to testing to product release, etc. Developers work on code changes in promotional states that are isolated from other development efforts.
promotion state	A state through which a product passes. For example, a software application goes though a development, test, and release cycle could use the promotion states Development, Test, and Release. In the application, each promotion state has a view label assigned to it. The view label can change over time, but testers, for example, always working in the Test state, can be oblivious to what label is currently assigned to that state.
promotion state configuration	The basis for a new view or a reconfigured view. The view contain only items with the promotion state you specify. This option is disabled in the new view's parent view or the reconfigured view has no promotion states defined for it.
properties	Attributes stored for each item (and each revision of an item) under version control.
protocol	A set of rules governing how something is done. Network protocols govern how data is transported over the network.
query	The criteria used to select a few items from among many. You can apply a query to items in the upper pane to display only the items that are of interest.

read-only reference view	A reference view that cannot be modified from the reference view—although it may be modified as the parent view is updated. If it floats, it is updated. If it is based on a label (or a promotion state) and the items with the specified label change, the read-only reference view with reflect that. It is based on a specific date and time, it is frozen as a copy of what the parent view looked like at that point in time.
reference count	A list of the items that reference another item. For example, a file may be shared by two project views on the same server (or even between folders in the same view) and, therefore, have two references to it.
reconfiguring	You can reconfigure a view, file, change request, topic, or task to a point in the past, defined by a label, promotion state, or a point in time.
reference view	A view derived from a parent view that, in general, uses a different application folder as the root folder and uses the same working folders at the parent. If it floats, it receives updates as the parent view changes. If it floats and is not read-only, it sends updates to the parent view as it changes. If the reference view is based a specific label or date and time within the parent view, it is frozen at that moment in time and is read-only.
refresh	The updating of the information in the project view window or only its upper pane. ■ Pressing *F5* to refresh the upper right pane for the current item. ■ Pressing *Ctrl+F5* to refresh the upper pane and simultaneously collapse all open groups ■ Pressing *Shift + F5* to refresh the entire view (all item lists in all the tabs as well as the folder hierarchy) ■ Pressing *F6* to refresh the files from an external archive, such as Visual SourceSafe or PVCS. Be sure to enable the All Descendants button (or select All Descendants from the File menu) before you press *F6* so that the refresh recurses through all the child folders.
remove from version control	To make an item invisible and inaccessible in the current view. However, all information about the item remains in the repository and, in the case of files, the archive. You can roll back a folder or view to a time when the item existed if you need to access it again
repository	Location in which the hive, attachment folder, log files, etc. are stored; associated with a specific server configuration.
repository customization	The Server feature that enables you to modify values of existing enumerated fields and create customized fields that represent information specific to your working environment.
Requirement component	The application enables users to create, track, and complete requirements related to the project. This component also imports requirements from CaliberRM.
response	Any of a number of replies to a topic that, along with the topic, form a hierarchical structure called a topic tree.
revision	As an item, such as a change request, is revised, each set of changes is saved as a revision.
revision label	A revision label provides a convenient method of identifying a revision of an item or a set of revisions by name. This is primarily used for files. For example, when you check in a group of files that may need to be checked out together, you can give them a revision label.
revision number	The revision number is an identification number assigned by the application to a revision of an item. It indicates how many consecutive revisions have been made since the item was originally created, perhaps in a different view.
root folder	The topmost folder in the folder hierarchy. The root folder in the root view of a project often has the same name as the view and the project.
secondary sort	Sorting items in a list that is already sorted (primary sort). For example a file list might be sorted by extension, then (in a secondary sort), sorted by name within groups of the same extension.

server	A computer or system that provides services to clients. The clients may be other computers.
server configuration	Contains the repository and option settings you select when you set up your Server. For example, the administrator may want projects to use encryption and compression, so the server configuration specifies these features. Application items, such as folders and files, can be shared provided their projects use the same server configuration. A server can be started with any one of several server configurations, but that configuration controls what projects, etc. can be accessed during that session.
shortcut	A file that starts another application, often with a specific document or set of data. The shortcut is usually stored as an icon on your desktop. For quick access, you can save shortcuts to project views, View Comparison windows, and individual items (except for files and audit entries).
SMTP	Simple Mail Transfer Protocol, commonly used for Internet and UNIX mail systems. It usually uses port 25. Sending items via e-mail in the application requires SMTP.
sort or primary sort	To place items in ascending or descending order in the upper pane of the project view window based on the value in one column. Depending on the values in the column, the values are sorted numerically, alphanumerically, or by an internal order or key. Click the column header once to sort. Click a second time to change the sort order from ascending to descending or vice versa. To group the sorted items by value, use the Sort and Group dialog.
StarDisk	StarDisk is a virtual file system that enables you to use conventional Windows applications, such as Windows Explorer, Microsoft Word for Windows, and Microsoft Developer Studio, to access and manage files that are under version control. You use StarDisk to map a view to a virtual drive. Then you can access any file on that drive from Explorer or another application. If the file is not checked out, StarDisk can check it out for you.
status	File status is the relationship between the working file and the tip revision in the repository. The file statuses are: In View, Not in View, Missing, Current, Merge, Modified, Out of Date, and Unknown. For a discussion of statuses, see “Determining the Status of Your Files” on page 130 and “Understanding the Effects of Status on Check-ins and Check-outs” on page 139
Task component	The application enables users to create, track, and resolve tasks related to the project. This component also interoperates with Microsoft Project.
TCP/IP	A protocol for communication between computers used by the Internet. Acronym for Transmission Control Protocol/Internet Protocol.
team member	Any of the project users. Sometimes used to denote a user that does not have administrator status.
test command	A command that you specify so that the application can perform, in proper sequence, an automated test procedure to test a defect.
text file or ASCII file	A file that contains only printable text characters, spaces, carriage returns, and sometimes tabs and an end-of-file marker, without any formatting codes. The application identifies as text files any files that contain no null characters. All other files are binary.
threaded conversation	A series of responses to a posted topic. Each conversation forms a topic tree with the topic as its root. Its called a threaded conversation because the tree hierarchy indicates whether a response is a reply to the topic or another response to that topic. By reading each response in a thread, one after the other, you can see how the discussion has evolved.
time stamp	Information maintained by the application about files and revisions. ■ For file revisions: The date and time that the file was checked into the application. ■ For files: The date and time for the working file.

tip revision	The most recent revision to an item such as a file or change request. The tip revision is based on your view's configuration. For example, if your view is configured to a particular label, the tip revision is the revision with that label. If your view is configured to a particular date and time, the tip revision is the one that was checked in just prior to that point in time.
topic	The first message on a particular subject attached to a folder in the folder hierarchy. After the topic is submitted by one team member, others may respond to it, creating a topic tree.
topic tree	One of perhaps many hierarchical structures that appear in the project view window when the Topic tab and the Tree Display button are selected. Its root is a topic and its branches consist of responses to that topic. A topic tree represents a threaded conversation.
Unknown	File status. The file in the working folder has the same name as a file in the view but the file was not checked out from the repository. You might have copied it from another location. Use Update Status to determine the correct status.
unlock	The process of releasing a locked file, indicating that you are no longer changing the file or considering changing it.
user	An individual given access to a server configuration and the projects it manages via the Server. Usually, that access is protected by password. A user is also referred to as a team member.
vault	Location in which revisions of the files under version control are stored. For StarTeam 6.0 and earlier releases, a Native-I vault. is used. For StarTeam 2005 and later releases, Native-II vaults are used. To understand the differences in vault architecture, see the *StarTeam Installation Manual* and the *StarTeam Administrator's Guide*.
variant view	See branching view.
version control	Version control is the process of storing and tracking the various changes (revisions) to one or more files. A version control system maintains the revision history generated as the files evolve into their final forms. The main advantage of using an automated version control system is a fast, easy recall of previous revisions. The application also tracks revisions of other items.
version control system	Application software to help manage multiple revisions of the same file.
view	Consists of a folder hierarchy, the items associated with each folder in that hierarchy, view properties, and view access rights. The behavior of the view may or may not permit branching, may or may not be read-only, and may or may not have a connection to its parent view.
view label	The main purpose of a view label is to "time stamp" the entire contents of the view. Then you can roll back the view to that label and see everything that existed at that point in time. However, unless the view label is frozen, you can make some adjustments. You can include or exclude some folders and items from the view label by attaching or detaching the view label. You can also move a view label from one revision to another. For example, a couple of files might not have been checked in prior to the creation of the label but need to be included. View labels are automatically and immediately attached to all folders and items in the view at the time they are created.
view window	The window in which the application displays an open view. Also called the project view window. It contains three panes. The left pane displays the folder hierarchy. The upper pane displays items associated with the folder selected from the folder hierarchy. The lower pane displays details, history and links for the item selected from the upper pane.
Visual Diff	A utility that compares two text files or two revisions of one text file and shows the differences (if any) between them.
Visual Merge	A utility that assists you as you merge two revisions of a file that are based on the same checked-in revision. It performs a three-way comparison, comparing the two revisions to each other and to their common ancestor.

work around	Steps needed to avoid a defect in a product. Work arounds are frequently offered until the revision that resolves a change request for the defect becomes available.
working file	Any file in a working folder can be considered a working file. Often, you have checked out the file for modification. When checked in, a working file becomes a revision. When checked in, a working file becomes a revision.
working folder.	Every folder has a corresponding working folder to which the working copies of files are checked out and from which files are added and checked in to the folder.

Index

Symbols

$Author$ keyword 144
$Date$ keyword 144
$Folder$ keyword 144
$Header$ keyword 144
Id keyword 144
$Locker$ keyword 144
Log keyword 144
$NoKeywords$ keyword 144
$Project$ keyword 144
$Revision$ keyword 144
$Workfile$ keyword 144
.sbas
  file statuses 278
?
  command-line option 287

A

About command (of Toolbar) 32
access rights
  defined 327
  filters 65
  labels 226
  queries 65
active
  command-line option 292, 300
  process items 212
Actual Finish task field 90
Actual Hours task field 90
Actual Start task field 91
add
  command at DOS prompt 292
Add Tool command (of Toolbar) 32
add-folder
  command at DOS prompt 293
adding
  files 127, 130
  files at command-line 292
  folders 44
  folders at command-line 293
  notes to tasks 178
  projects at command-line 295
  task attachments 179
  tasks to projects 174
  tools to Toolbar 35
  views at command-line 297
adding files
  defined 327
add-project
  command at DOS prompt 295
Addressed By
  column in change request list 159
Addressed By change request field 76
Addressed In
  column in change request list 159
Addressed In change request field 76
Addressed In View change request field 76
add-view
  command at DOS prompt 297
administrators
  defined 327
advanced fields 65
all descendants button 131
All Descendants buttons
  defined 327
alphanumeric
  defined 327
alternate working folders 24
  defined 327
Am I Recipient? topic field 98
Am I Responsible? requirement field 84
am option
  Visual Merge 114
Ambiguities Found requirement field 84
amt option
  Visual Merge 114
ANDing 56
application
  client refresh 274
apply-label
  command at DOS prompt 298
archive
  defined 327
Archive Path file field 67
Archive Type file field 67
as of
  configurations 23, 201
  revisions 201
ASCII files
  comparing 111
  defined 327, 334
  merging 111
assigning
  change requests 163, 166
  resources 176
at option
  Visual Merge 114
attaching
  labels 224
  labels to folders 223
Attachment Count change request field 76
Attachment Count requirement field 84
Attachment Count task field 91
Attachment Count topic field 98
Attachment IDs change request field 76
Attachment IDs requirement field 84
Attachment IDs task field 91
Attachment IDs topic field 98
Attachment Names change request field 77
Attachment Names requirement field 84
Attachment Names task field 91
Attachment Names topic field 98
attachments
  change requests 162
  tasks 178
  topics 184
Attention Notes task field 91
audit entries
  charts 259
  defined 327
  finding 205
  operations on 191
  printing 206

refreshing 188
searching fields for specific data 205
audit log
columns 187
controlling 187
defined 327
displaying 187
double-clicking 188
events 189
filters 188
menus 188
Show All 188
sorting 54, 187
Type 187
User 187
audit log fields 103
Audit Trail tabs 187
Author
defined 327
history list 203
topics 183
Auto
StarTeamMPX status 283
autoLogon
bco command option 319
automatic log on 34
overriding 34
automatic refresh
defined 327

B

b
command-line option 306, 309
ba
command-line option 297
bars, splitter 121
bco command
autoLogon option 319
Cache Agent use 324
cfgd option 320
cfgl option 320
cfgp option 320
checking out files regardless of status 322
cmp option 320
dryrun 320
encrypt option 320
end-of-line conversion 320
eol option 320
file statuses 322
files specification 321
filter option 322
folder option 323
forcing check-outs 322
fs option 322
h option 322
help option 322
including child folders 322
including folders 322
is option 322
netmon option 322
o option 322
overriding specific folder's working folder 322
overriding view's working folder 324
p option 323
password files 324
preview 320
pwdfile option 324
read-only files 324
ro option 324
rp option 322, 324
SDK NetMonitor information 322
t opton 324
time stamps 324
timing statistics 324
ts option 324
use of Toolbar 319
useCA option 324
vb option 324
verbose 324
view configuration as of date 320
view configuration label names 320
view configuration states 320
volume statistics 324
bulk check-out utility
See also bco command
bco command line 319
behavior
branching 197
branching disabled 197
branching not set 198
branching set 198
change requests 198
files 198
folders 198
non-branching 197
shared items 195
benefits 3, 13
requirement list 151
requirement trees 151
task list 173
task trees 172
topic list 183
topic trees 182
binary files
defined 328
differences 305
merging 118
bitmaps
copying charts as 264
bits
executable 132
black
matching text color 115, 121
blank branching views
defined 328
blue
changed text color 115, 122
bn
command-line option 297
bold
change requests 160
requirements 151
tasks 173
bold convention 2
Boolean
relational operators 66
Bpvcs
command-line option 306
brackets convention 2
Branch On Change change request field 77
Branch On Change file field 67

Branch Revision
history list 204
Branch State change request field 77
Branch State file field 68
branching
change requests 198
defined 328
disabled 197
files 198
folders 198
not set 198
set 198
branching behavior
defined 197
branching views
at command-line 297
defined 328
break
command-line option 310
breaking
locks 202
locks at command line 310
build labels 216
command line 309
defined 328
type of view labels 217
builds
defined 328
effect on change request verification 167
labels 128
bulk check-out utility 319
buttons
chart toolbar 264
By Status and Responsibility
change requests 160

C

c
command line option 306
Cache Agent
bco command use 324
cached server/user associations 33
caching user name and password (with Toolbar) 34
CaliberRM
requirements 154
CaliberRM Import DocFactory
configuring 153
case
ignoring 124
case-sensitivity settings
folders 26
Category change request field 77
cfgd
bco command option 320
command-line option 297, 302, 305, 306, 308, 311, 312, 313, 317
cfgl
bco command option 320
command-line option 297, 302, 305, 306, 308, 311, 312, 313, 317
cfgp
bco command 320
command-line option 297, 302, 305, 306, 308, 311, 313, 317
change request
charts 168
change request fields 76
change request list
Addressed By 159
Addressed In 159
Created By 159
Created Time 159
Description 159
Fix 159
Last Build Tested 159
Modified Time 159
Priority 159
sorting 54, 159
Test Command 159
Work Around 159
change request numbers
defined 328
change requests
access rights for labels 226
assigning 163, 166
at command line 292, 300
attaching labels 224
attachments 162
behavior 198
bold 160
branching 198
build labels 216
build's effect on 167
By Status and Responsibility 160
charts 259
closing 165, 168
comparing properties 203
comparing revisions 203
configuring 200
copying revision labels 220
creating revision labels 224
custom fields 162
defect type 161
deleting 209
description 162
detaching labels 225
details 161
double-clicking 158
drag-and-drop 160, 164
filters 160
finding 205
fixed 211
fixing 163
flagging 208
history 161
icons 158
label overview 215
linking 161, 227
marking read or unread 208
menus 158
merging 205
moving 193
moving labels 224
Not a Priority 160
notifications 14, 281
opening 163, 165
operations on 191
overview 13, 155
printing 206
Priority 160
process items 211

read 160
refreshing 158
reopening 165
reports 168
reports available 253
resolving 163, 167
responsibility 162
reviewing 163
rolling back 200
searching fields for specific data 205
selecting by label 225
selecting by query 56
sending e-mail 206
severity 161
sharing 195
shortcuts 194
Show All 160
Show Unread Changes 160
statuses 157, 160
submitting 161, 166
suggestion type 161
synopsis 162
testing 165
Type 160
verifying 164, 168
workflow affected by change in view 193
workstation features 280
change requests list
defined 328
changed text color 115, 122
changing
change request behavior 198
chart colors 264, 265
chart data 265
chart fonts 267
chart legends 266
chart patterns 265
chart titles 264, 267
chart types 264
file behavior 198
folder behavior 198
link properties 229
passwords 283
Visual Diff colors 122
charts
adding depth 264
audit entries 259
available 259
change requests 168, 259
changing colors 264, 265
changing data 265
changing patterns 265
changing types 264
copying as bitmaps 264
copying as data 264
correlation 263
creating 259
distribution 262
editing legend text 266
editing title text 264, 267
exploding pie 267
exporting 264
files 259
fonts 267
grids 264
importing 264
legends 264
options 264
palette bar 266
pattern bar 266
printing 264
removing patterns 266
rotating 264, 267
saving 264
simple 260, 262
swiveling 267
tasks 259
tilting 267
time-series 263
toolbar 264
tools 264
topics 259
two/three dimensional 264, 267
window 264
z-cluster 264
zooming 264
check-in
defined 328
checking in
effects of statuses 139
file revisions 134
files 130
files as unlocked 276
files at command-line 299
forced at command line 300
forced at command line interactively 300
modified files at command line 300
reasons at command line 301
revisions 127
checking out
effects of statuses 139
files 130, 136, 319
files as locked 276
files at command line 302
files at command-line 301
files in bco command 322
files over slow connection 275
forced at command line 304
forced at command line interactively 303
forced in bco command 322
read-only files 276
revisions 127, 136
time stamps 275
unlocked files 276
check-out
defined 328
checksums
determining status 277
child folders
bco command option 322
command-line 289
child requirements
creating 152
Children Count requirement field 85
Children Count task field 91
Children Count topic field 98
ChildType requirement field 85
ci
command at DOS prompt 299
Class Name 1 audit log field 103
Class Name 2 audit log field 103
Class Name 3 audit log field 103

clearing
flags 208
client-side
statuses 24, 277
cloning
revision labels 219, 220
Closed
task status 175
Closed On change request field 77
closing
change requests 165, 168
project view windows 27
Toolbar window 38
cmp
bco command option 320
command-line option 287
co
command at DOS prompt 301
codes
at command line 306
changing colors 122
colors in Visual Diff 121
collapsing
folder hierarchy 8
requirement trees 151
task trees 172
Toolbar window 38
topic trees 182
colors
changed text 115, 122
changing chart fonts 267
changing charts 265
changing Visual Diff 122
charts 264
deleted text 115, 122
inserted text 115, 122
matching text 115, 121
Visual Diff 121
Visual Merge 118
column headers
defined 329
columns
audit log 187
history list 203
option at command line 307
sorting data 9
command-line
active option 292, 300
ba option 297
bn option 297
cr option 292, 300
d option 292, 293, 295, 298, 308, 309
dr option 298
ex option 294, 295
exfile option 294, 295
exlist option 294, 296
kw option 296
kwfile option 296
mark option 292, 300
name option 294, 296, 298
req option 292, 301
ro option 298
s option 296
task option 292, 301
command-line files
defined 329
command-line interface 285
? option 287
adding files 292
adding folders 293
adding projects 295
adding views 297
applying labels 298
b option 306, 309
Bpvcs option 306
break option 310
breaking locks 310
build labels 309
bypassing error messages 291
c option 306
case 306
cfgd option 297, 302, 305, 306, 308, 311, 312, 313, 317
cfgl option 297, 302, 305, 306, 308, 311, 312, 313, 317
cfgp option 297, 302, 305, 306, 308, 311, 313, 317
changing file descriptions 307
checking in files 299
checking in modified files 300
checking out files 301
checking out files regardless of status 302
checking out files that are missing or out-of-date 302
cmp option 287
comparing files 305
copyright notices 289
creating labels 309
creating working folders 312
deleting files 304, 314
deleting working files 314
derived views 298
df option 314
displaying file histories 308
e option 306
encrypt option 287
end-of-line conversion 292, 300, 302, 306, 317
eol option 292, 300, 302, 306, 317
exit codes 306
f NCI option 300
f NCO option 302
f option 309
file descriptions 292, 308, 309
file statuses 299, 300, 302, 305, 306, 308, 310, 311, 313, 314, 317
files specification 288
filter option 299, 300, 302, 305, 306, 308, 310, 311, 313, 314, 317
folder descriptions 293
folder exclude lists 294, 295, 296
folder names 294
folder option 290
forcing check-ins 300
forcing check-ins interactively 300
forcing check-outs 304
forcing check-outs interactively 303
fs option 302
h option 287
halting execution for error 291
help 287
help option 287
hook option 303
i option 300, 303, 306

ignoring characters in specific columns 307
including child folders 289
including folders 289
is option 289
keyword extensions 296
l option 292, 300, 303, 310
labels 293, 299, 300, 301, 304, 307, 309
labels as of date 309
labels based on promotion state 310
labels copied 310
labels frozen 309
lbl option 299
line number suppression 307
listing child folders 311
listing files 311
locking files 292, 300, 303, 310
locking servers 314, 315
m option 307
merge option 303
mode option 315
moving file statuses 312
n option 307
nd option 307
nel option 292, 300, 304, 310
nl option 309
nologo option 289
nomove option 300
non-exclusive locking of files 292, 300, 304, 310
o option 300, 304
overriding view's working folder 289, 290
p option 290
password files 290
process items 292, 300, 301
project descriptions 295, 298
project names 296
pwdfile option 290
q option 290
r option 301, 309
read-only files 292, 301, 304, 310
read-only view 298
read-write files 292, 301, 304, 310
reason for check-in 301
removing files 314
revision labels 309
revision numbers 299, 304, 307
revisions as of date 299, 304, 307
rf option 301
ro option 292, 301, 304, 310
rp option 289, 290
rw option 292, 301, 304, 310
s option 315
Server 296
setting personal options 315
sf option 311
specifying servers 315
stcmd add 292
stcmd add-folder 293
stcmd add-project 295
stcmd add-view 297
stcmd apply-label 298
stcmd ci 299
stcmd co 301
stcmd delete-local 304
stcmd diff 305
stcmd dsc 307
stcmd hist 308
stcmd label 309
stcmd lck 310
stcmd list 311
stcmd local-mkdir 312
stcmd remove 314
stcmd server-mode 314
stcmd update-status 316
stop option 291
suppressing differences 307
suppressing progress reports 290
syntax 286
t option 307
tab stops 307
time stamps 304
ts option 304
u option 293, 301, 304, 310
unlocking files 293, 301, 304, 310
unlocking servers 314, 315
updating file statuses 316
vd option 299, 304, 307, 309
view configuration as of date 297, 302, 305, 306, 308, 311, 312, 313, 317
view configuration label names 297, 302, 305, 306, 308, 311, 312, 313, 317
view configuration states 297, 302, 305, 306, 308, 311, 313, 317
view names 298
view types 297
vl option 293, 299, 301, 304, 307, 310
vn option 299, 304, 307
vp option 310
w option 307
whitespace 306, 307
x option 291
command-line option 290
commands
viewmerge 114
comma-separated data 251
Comment
link list column 229
Comment change request field 78
Comment file field 68
Comment requirement field 85
Comment task field 91
Comment topic field 98
CommentID change request field 78
CommentID file field 69
CommentID requirement field 85
CommentID task field 91
CommentID topic field 99
comments
editing for files 204
task revisions 179
Comments requirement field 85
common files
See sharing files.
comparing
change request properties 203
change request revisions 203
file properties 203
file revisions 140
files 111, 140
files at command-line 305
revisions 111, 140
revisions and working files 140, 141
task properties 203

task revisions 203
three ways 111
topic properties 203
topic revisions 203
two working files 141
two-way difference 111
versions 111
Visual Diff and Visual Merge 111
comparison
color code 121
comparison utilities
default 280
Component change request field 78
components
defined 329
compressing
data 21
data at command line 287
data in bco command 320
compression
defined 329
disabling 21
enabling 21
shortcuts 21
Compression Level file field 69
conditions
creating 57, 59
Configuration Time change request field 78
Configuration Time file field 69
Configuration Time requirement field 85
Configuration Time task field 91
Configuration Time topic field 99
configurations
as of 23, 201
as of date at command line 297, 302, 305, 306, 308, 311, 312, 313, 317
as of date in bco command 320
date/time 23, 201
defined 329
floating 201
labeled 23
labeled revisions 201
promotion state 23
promotion states 201
configuring
CaliberRM Import DocFactory 153
change requests 200
files 200
folders 200
tasks 200
topics 200
confirming
deletions 273
move operations 273
sharing operations 273
warnings 273
conflicts
Visual Merge 117
connection properties 27
Constraint Date task field 92
Constraint Type task field 92
containers
defined 329
Content 260
Content Revision file field 69
Content topic field 99
contents
topics and responses 188
controlling
task lists 171
task trees 171
upper pane 8
conventions
bold 2
brackets 2
fixed-space type 2
italics 2
menu selections 2
vertical bars 2
conversions
end-of-line 292, 300, 302, 306, 317, 320
eol 292, 300, 302, 306, 317, 320
copying
chart bitmaps 264
chart data 264
filters 64
queries 60
revision labels 219, 220
view labels 217
copyright notices
suppressed at command line 289
correlation charts 263
cr
command-line option 292, 300
CR Number change request field 78
Created By
column in change request list 159
column in requirement list 150
link list column 229
Created By audit log field 103
Created By change request field 78
Created By file field 69
Created By requirement field 85
Created By task field 92
Created By topic field 99
Created On
link list column 229
Created Time
column in change request list 159
column in requirement list 150
Created Time audit log field 103
Created Time change request field 78
Created Time file field 69
Created Time requirement field 85
Created Time task field 92
Created Time topic field 99
creating
change requests 161
charts 259
conditions 57, 59
filters from scratch 62
filters using current settings 62
folder structures 43
folders 44
labels at command-line 309
queries 56
requirements 152
responses 185
revision labels 219
simple charts 262
tasks 174
topics 184

view labels 217
work records for tasks 177
working folders 43
Current
file status 139
custom fields
change requests 162
files 147
tasks 178
topics 184
customization
defined 333
customizing
Toolbar 35
Visual Merge 118

D

d
command-line option 292, 293, 295, 298, 308, 309
data
changing charts 265
compressing 21
compressing at command line 287
compressing in bco command 320
copying charts 264
encrypting 21
encrypting at command line 287
encrypting in bco command 320
Date Range
topics 183
date/time
configurations 23, 201
relational operators 66
revisions 201
defect
change request type 161
defect tracking
defined 329
defects
defined 329
Delete Tool command (of Toolbar) 32
Deleted By audit log field 103
Deleted By change request field 78
Deleted By file field 69
Deleted By requirement field 85
Deleted By task field 92
Deleted By topic field 99
deleted text color 115, 122
Deleted Time audit log field 103
Deleted Time change request field 79
Deleted Time file field 70
Deleted Time requirement field 86
Deleted Time task field 92
Deleted Time topic field 99
delete-local
command at DOS prompt 304
deleting
change requests 209
defined 329
files 209
files at command line 314
files at command-line 304
filters 65
folders 209
labels from view 221
queries 60
tasks 209
tools from Toolbar 37
topics 209
work records in tasks 177
working files at command line 314
deletions
confirming 273
delta storage
defined 330
depth
charts 264
derived views
at command-line 298
Description
column in change request list 159
column in requirement list 150
Description change request field 79
Description file field 70
Description requirement field 86
descriptions
change requests 162
changing folder descriptions 46
files at command-line 292, 308, 309
folders at command-line 293
projects at command-line 295
topics and responses 188
views at command-line 298
desktop
shortcuts 22
detaching
labels 225
labels from folder and contents 223
detail list 10
detail reports
printing 250
Detail tab
change requests 161
description 10
files 129, 188
requirements 152
tasks 172
topics 182
developer support 1
df
command-line option 314
diff
command at DOS prompt 305
differences 116, 122
at command line 305
color codes 121
suppressed at command line 307
Visual Diff 122
Visual Merge 117
See comparing.
directories
See folders.
Disabled requirement field 86
disabling
branching 197
disconnecting
from server 27
displaying
audit log 187
chart legends 264
file list 128

history list 203
task trees 171
Toolbar About dialog 39
Toolbar help 38
Toolbar window 31
Toolbar window at startup 31
distribution charts 262
do not level 176
DOS prompt
commands for 285
Dot Notation change request field 79
Dot Notation file field 70
Dot Notation ID change request field 79
Dot Notation ID file field 70
Dot Notation requirement field 86
Dot Notation task field 92
Dot Notation topic field 99
double-clicking 129
audit log 188
change requests 158
requirements 149
responses 182
tasks 172
topics 182
dr
command-line option 298
drag-and-drop
change requests 160, 164
files 129
folders 42
requirements 150
tasks 172
dryrun
bco command option 320
dsc
command at DOS prompt 307

E

e
command-line option 306
editing
chart legends 266
chart title text 264
chart titles 264, 267
charts 265
files 133, 136
filters 63
queries 60
report templates 255
repository customization 147
revision comments for files 204
work records in tasks 177
editors
default 279
e-mail
defined 330
sending change requests 206
sending items as shortcuts 207
sending tasks 206
sending topics 206
enabling
compression 21
encryption 21
encoding
files 115, 120
encrypt
bco command option 320
command-line option 287
encrypting
data 21
data at command line 287
data in bco command 320
encryption
defined 330
disabling 21
enabling 21
shortcuts 21
End Modified Time change request field 79
End Modified Time file field 70
End Modified Time requirement field 86
End Modified Time task field 92
End Modified Time topic field 99
end-of-line markers
ignoring 125
endpoints
defined 330
Entered By change request field 79
Entered On change request field 79
enumerated type
relational operators 66
envelope 183
EOL
ignoring 125
options for Cross-Platform Client 277
eol
bco command option 320
command-line option 292, 300, 302, 306, 317
EOL Character file field 70
EOL settings
folders 26
error messages
command-line 291
errors
halting execution of command-line 291
Estimated Finish task field 93
Estimated Finish Variance task field 93
Estimated Hours task field 93
Estimated Hours Variance task field 93
Estimated Start task field 93
Estimated Start Variance task field 93
Event audit log field 103
events
audit log 189
ex
command-line option 294, 295
exclude lists
folders at command-line 294, 295, 296
exclusive lock 131
exclusive locks
defined 330
executable bit 132
Executable file field 70
exfile
command-line option 294, 295
exit codes 291
at command line 306
Exit command (of Toolbar) 32
exiting
Toolbar 39
Visual Merge 117
exlist

command-line option 294, 296
expanding
folder hierarchy 8
requirement trees 151
task trees 172
Toolbar window 38
topic trees 182
Expected Effort requirement field 86
exploding
pie charts 267
exporting
charts 264
data for use in other applications 251
reports 251
Extension file field 70
extensions
keywords at command line 296
External Reference change request field 79
External Reference requirement field 86

F

f
command-line option 309
f NCI
command-line option 300
f NCO
command-line option 302
features
change requests 280
files 275
tasks 280
topics 280
fields
advanced 65
common 65
file 129
file
command-line descriptions 307
file compression
defined 330
file fields 67
file list
displaying 128
sorting 54
file statuses
Current 139
effects on checking files in and out 139
Merge 139
Missing 139
Modified 139
Not In View 139
Out Of Date 139
purging 278
storing 24
Unknown 139
File Time Stamp at Check-In file field 71
File Type file field 71
files
access rights for labels 226
adding 127
adding to project 130
affected by change in view 193
attaching labels 224
behavior 198
branching 198
charts 259
charts available 259
check-in reasons at command line 301
checking in 127, 134
checking in and unlocking 276
checking in modified files at command line 300
checking out 127, 136, 319
checking out at command line 302
checking out in bco command 322
command-line specification 288
comma-separated text files 251
comparing 111, 140
comparing properties 203
comparing two revisions 140
comparing working file to latest revision 141
comparing working file to revision 140
configuring 200
controlling the list 128
copying revision labels 220
creating revision labels 224
custom fields 147
deleting 209
deleting at command line 314
deleting working files at command line 314
descriptions at command-line 292, 308, 309
detaching labels 225
details 129, 188
detecting changes to working files 277
displaying file histories 308
drag-and-drop 129
editing 133
editing revision comments 204
effects of statuses on checking files in and out 139
encoding 115, 120
filters list 129
finding 205
finding matches 116, 122
flagging 208
forcing check-ins at command line 300
forcing check-ins at command line interactively 300
forcing check-outs at command line 304
forcing check-outs at command line interactively 303
forcing check-outs in bco command 322
history 129
in view 130
keywords 144
label overview 215
linking 129, 227
linking to process items 213
listing at command line 311
locked from integration 276
locking 135
locking at command line 310
locking at command-line 292, 300, 303, 310
locking on check-out 276
managing 127
merging 111, 205
merging binary files 118
merging compared 123
merging versions 127
moving 193
moving labels 224
moving statuses at command line 312, 315
non-excluded from view 130
not in view 130, 131

opening 133
operations on 191
optimizing check-outs over slow connection 275
order in Visual Diff 121
passwords at command-line 290
passwords in bco command 324
personal options 147
printing compared 123
printing file properties 206
properties 145
read-only 276
read-only command-line 292, 301, 304, 310
read-only in bco command 324
read-write command-line 292, 301, 304, 310
refreshing 129
removing at command line 314
removing from version control 209
reports available 253
reviewing contents of revisions 140
rolling back 200
searching fields for specific data 205
selecting by label 225
selecting by query 56
setting personal options 275
sharing 195
sorted by status 130
sorting list 129
specification in bco command 321
status repository options 24, 277
statuses at command line 299, 300, 302, 305, 306, 308, 310, 311, 313, 314, 317
statuses determined by checksums 277
statuses in bco command 322
three-way merge 111
time stamping at command line 304
time stamping in bco command 324
time stamps of checked out files 275
to be checked in 130
to be checked out 130
two-way difference 111
unlocking 135
unlocking at command line 310
unlocking at command-line 293, 301, 304, 310
updating status 134
using Visual Merge as a stand-alone utility 113
workstation features 275
Files tab 128
-filter 288
filtering
audit log 188
change requests 160
file list 129
requirements 151
tasks 173
topics 183
filters
access rights 65
bco command 322
command-line option 299, 300, 302, 305, 306, 308, 310, 311, 313, 314, 317
copying 64
creating from scratch 62
creating with current settings 62
defined 330
deleting 65
editing 63
files 130
performance 53
resetting 64
upper pane 61
using 62
finding
audit entries based on field content 205
change requests based on field content 205
differences in compared files 116, 122
files based on field content 205
matches in compared files 116, 122
tasks based on field content 205
text in compared files 116, 122
topics based on field content 205
Finish
task status 175
Fix
column in change request list 159
Fix change request field 80
fixed
change requests 330
links 229
fixed-space type convention 2
fixing
change requests 163
Flag change request field 80
Flag file field 71
Flag requirement field 86
Flag task field 93
Flag topic field 100
Flag User List change request field 80
Flag User List file field 71
Flag User List requirement field 87
Flag User List task field 93
Flag User List topic field 100
Flagged Items
requirements 151
flagging
change requests 208
files 208
items 208
tasks 208
topics 208
flags 183
creating 208
removing 208
floating
configurations 201
defined 330
links 229
Folder
link list column 229
folder
defined 330
Folder audit log field 104
Folder change request field 80
Folder file field 71
folder hierarchy 42
collapsing 8
defined 330
expanding 8
Folder Path audit log field 104
Folder Path change request field 80
Folder Path file field 71
Folder Path requirement field 87
Folder Path task field 93

Folder Path topic field 100
Folder VMID audit log field 104
folders
  access rights for labels 226
  adding to project view 44
  affected by change in view 193
  alternate working folders 44, 48
  attaching labels 223
  bco command 322
  bco command option 323
  behavior 198
  branching 198
  case-sensitivity settings 26
  changing working folders 48
  command-line 289
  command-line option 290
  configuring 200
  creating 44
  creating revision labels 222
  creating structures 43
  creating working folders at command line 312
  deleting 209
  descriptions 46
  descriptions at command-line 293
  detaching labels 223
  drag-and-drop 42
  EOL settings 26
  exclude lists at command-line 294, 295, 296
  icons 8
  invisible 25
  label overview 215
  labeling 222
  linking 227
  listing child folders at command line 311
  merging 205
  moving 42, 193
  moving change requests 193
  moving labels 222
  moving tasks 193
  moving topics 193
  names 46
  names at command-line 294
  operations on 191
  overriding working folder at command line 289, 290
  overriding working folder in bco command 322, 324
  overview 41
  profiles 25
  properties 46
  references 236
  reset scope 273
  restoring selection 273
  rolling back 200
  sharing 195
  sharing change requests 195
  sharing files 195
  sharing tasks 195
  sharing topics 195
  visible 25
  working 42, 43
font
  charts 267
  Visual Merge 118
foreign refresh 27, 129
  defined 330
freezing
  change requests 200
  files 200
  folders 200
  labels 220
  tasks 200
  topics 200
frozen
  defined 330
  labels 220
fs
  bco command option 322
  command-line option 302
full revision storage
  defined 330

G

gallery
  changing chart data 265
  changing chart types 264
green
  inserted text color 115, 122
grids
  charts 264
Grouped by Creator
  requirements 151
Grouped by Status
  requirements 151
grouping
  versions by label 128

H

h
  bco command option 322
  command-line option 287
halting
  errors at command-line 291
help
  bco command option 322
  command-line option 287
  for Toolbar 38
  status bar 12
Help command (of Toolbar) 32
hiding
  chart legends 264
  taskbar button (for Toolbar window) 32
  Toolbar system tray icon 32
High Effort requirement field 87
hist
  command at DOS prompt 308
history
  change requests 161
  displaying file histories 308
  files 129
  requirements 152
  tasks 172
  topics 182
history list 10
  Author 203
  Branch Revision 204
  columns 203
  defined 330
  displaying 203
  Reason for Check-In 204
  reports available 253
  Revision 203

Time Stamp 203
View 203
history pane
local times 273
UTC times 273
History tab 203
change requests 161
description 10
files 129
requirements 152
tasks 172
topics 182
Hold
task status 175
hook
command-line option 303
horizontal
chart grids 264

I

i
command-line option 300, 303, 306
I Am Responsible
requirements 151
icons
change requests 158
folders 8
requirements 151
icons 151
task trees 172
ignoring
case 124
end-of-line markers 125
whitespace 124
importing
charts 264
In Progress
task status 175
initial references 238
inserted text color 115, 122
Instant
StarTeamMPX status 282
integrations
locked files 276
interface
command-line 285
introducing
change requests 13
tasks 3
topics 16
is
bco command option 322
command-line option 289
Is My Task? task field 93
Is Replicated task field 94
italics convention 2
Item
link list column 229
Item 1 audit log field 104
Item 1 Info audit log field 104
Item 2 audit log field 104
Item 2 Info audit log field 104
Item 3 audit log field 104
Item 3 Info audit log field 105
Item Details
link list column 229
Item Type
link list column 229
Item Version
link list column 229
items
access rights for labels 226
affected by change in view 193
as shortcuts via e-mail 207
confirming move operations 273
confirming share operations 273
defined 330
flagging 208
linking 227
locking 202, 282
managing 191
merging 205
operations on 191
printing 206
process 211
read 208
references 236
shortcuts 194
unlocking 202, 282
unread 208
iw option
Visual Merge 114

K

keep current lock status
defined 330
key, merged files 123
keyword expansion
defined 331
keywords
$Author$ 144
$Dater$ 144
$Folder$ 144
$Header$ 144
Id 144
$Locker$ 144
Log 144
$NoKeywords$ 144
$Project$ 144
$Revision$ 144
$Workfile$ 144
defined 331
extensions for projects at command-line 296
files 144
using 144
kw
command-line option 296
kwfile
command-line option 296

L

l
command-line option 292, 300, 303, 310
Label tab
change requests 161
description 10
files 129
requirements 152
tasks 172

topics 182
label tree 10
labeled configurations 23
labeling
folder and its contents 222
labels
access rights 226
applying at command-line 298
as of date (at command line) 309
attaching 224
attaching to folders 223
based on promotion state (at command line) 310
build 128
change requests 215
command at DOS prompt 309
command line 309
command-line 293, 299, 300, 301, 304, 307
copied (at command line) 310
copying revision labels 220
copying view labels 217
creating 217, 219
creating and attaching revision labels to a folder revision 222
creating and attaching revision labels to an item revision 224
deleting from view 221
detaching 225
detaching from folders 223
files 215
folders 215
freezing and unfreezing 220
frozen (at command line) 309
grouping versions 128
in branching views 193
Label tab 10
milestone 128
moving from one folder revision to another 222
moving from one item revision to another 224
not reusable 128
overview 215
properties 221
selecting 225
tasks 215
topics 215
Last Build Tested
column in change request list 159
Last Build Tested change request field 80
Last MS Project Update task field 94
Last Work/ Dependency Update task field 94
launching tools (from Toolbar) 35
lbl
command-line option 299
lck
command at DOS prompt 310
left panes 7
legends
editing charts 266
light bulb 183
line number suppression
option at command line 307
line numbers
Visual Merge 118
lines
suppressing numbers at command line 307
link list 10
sorting 54
link properties
changing 229
Link tab
change requests 161
description 11
files 129
requirements 152
tasks 172
topics 182
linking
change requests 161
file revisions to process items 213
files 129
fixed 229
floating 229
item properties 231
items 227
link properties 231
linked items' properties 231
requirements 152
tasks 172
to specific revisions 229
topics 182
links
columns in link list 228
Comment column 229
Created By column 229
Created On column 229
Folder column 229
Item column 229
Item Details column 229
Item Type column 229
Item Version column 229
Link tab 11
process items 213
properties 231
Selection Version column 229
View column 229
list
command at DOS prompt 311
List Display button 151, 173, 183
listing files
at command-line 311
local-mkdir
command at DOS prompt 312
locating
differences in compared files 116, 122
matches in compared files 116, 122
text in compared files 116, 122
lock status 131
Locked By
column in requirement list 150
Locked By change request field 80
Locked By file field 71
Locked By requirement field 87
Locked By task field 94
Locked By topic field 100
locking
files 135
files at command-line 292, 300, 303, 310
files used from integrations 276
files when checked out 276
items 202, 282
servers at command-line 314
locks
breaking 202

breaking at command line 310
log
See audit log.
logged on user 27
logging off 27
logging on
as default user 33
as different user 34
automatically 34
bco command 319
logical operators 56
logon time 27
logs
output paths 274
StarTeam.log 274
Low Effort requirement field 87
lower pane
controlling 10
explained 7

M

m
command-line option 307
Maintain Group State on Folder/Scope Change 274
managing
files 127
items 191
tasks 171
Toolbar window 37
topics 181
versions 127
Manual
StarTeamMPX status 283
manual
overview 4
MAPI
defined 331
mark
command-line option 292, 300
marking
change requests 208
tasks 208
topics 208
matches
finding 116, 122
Visual Diff 122
matching text color 115, 121
MD5
determining status 277
MD5 Checksum file field 71
MD5 checksums
status determined by 134
menu commands
Visual Diff 119
menu selections convention 2
menus
audit log 188
change requests 158
help in status bar 12
tasks 172
Merge
defined 331
file status 139
merge
command-line option 303
merge utilities
default 279
merging
at command line 303
binary files 118
change requests 205
compared files 123
defined 331
file versions 127
files 205
files outside the application 113
folders 205
items 205
revisions 111
tasks 205
third-party application 303
three ways 111
topics 205
Message Broker 19, 20
messages
errors at command-line 291
Milestone task field 94
milestones 175
defined 331
labels 128
Missing
defined 331
file status 139
mode
command-line option 315
Modified
defined 331
file status 139
Modified By audit log field 105
Modified By change request field 80
Modified By file field 72
Modified By requirement field 87
Modified By task field 94
Modified By topic field 100
Modified Time
column in change request list 159
Modified Time audit log field 105
Modified Time change request field 80
Modified Time file field 72
Modified Time requirement field 87
Modified Time task field 94
Modified Time topic field 100
Modify Tool command (of Toolbar) 32
modifying
file properties 145
program shortcut properties 36
task properties 179
topic properties 185
view properties 26
See changing.
moving
change requests 193
confirming operations 273
file statuses at command-line 312
files 193
folders 193
tasks 193
topics 193
MS Project File Name task field 94
MS Task GUID task field 94
MS Task Unique ID task field 95

MS WBS Code task field 95
Multicast Service 19, 20
my account 283
My Lock change request field 81
My Lock file field 72
My Lock requirement field 87
My Lock task field 95
My Lock topic field 100

N

n
  command-line option 307
Name
  column in requirement list 150
name
  command-line option 294, 296, 298
Name file field 72
Name requirement field 88
names
  changing folder names 46
  folders at command-line 294
  projects at command-line 296
  views at command-line 298
NCI
  command-line option 300
NCO
  command-line option 302
nd
  command-line option 307
needs attention 176
Needs Attention task field 95
nel
  command-line option 292, 300, 304, 310
netmon
  bco command option 322
New Folder wizard 44
New Revision Comment change request field 81
New Revision Comment file field 72
New Revision Comment requirement field 88
New Revision Comment task field 95
New Revision Comment topic field 100
nl
  command-line option 309
nologo
  command-line option 289
nomove
  command-line option 300
non-branching behavior
  defined 197
non-exclusive lock 131
  defined 331
Non-Exclusive Lockers change request field 81
Non-Exclusive Lockers file field 72
Non-Exclusive Lockers requirement field 88
Non-Exclusive Lockers task field 95
Non-Exclusive Lockers topic field 100
Not a Priority
  change requests 160
Not In View
  file status 139
Not in View
  defined 331
not in view
  file status 131
Not Started
  task status 175
Notepad
  default editor 279
Notes
  added to tasks 178
Notes requirement field 88
Notes task field 95
notification
  system tray 281
notification of events
  defined 332
notifications
  change requests 14, 281
  tasks 14, 281
  topics 14, 281
Number
  column in requirement list 150
Number requirement field 88
numbers
  suppressed at command line 307
numeric
  defined 332
  relational operators 66

O

o
  bco command option 322
  command-line option 300, 304
object
  defined 332
Object ID audit log field 105
Object ID change request field 81
Object ID file field 72
Object ID requirement field 88
Object ID task field 95
Object ID topic field 101
off
  logging 27
Open command (of Toolbar) 32
Open Tools Folder command (of Toolbar) 32
opening
  change requests 163, 165
  files 133
  projects 20
  Toolbar window 30
operations
  folders 191
  items 191
operators
  logical 56
  relative 56
options
  charts 264
  file status storage 24, 277
  setting file 275
  setting personal options at command-line 315
  Visual Merge 276
order of
  files in Visual Diff 121
ORing 56
Out Of Date
  file status 139
Out of Date
  defined 332
output paths

reports 274
StarTeam.log 274
overriding
automatic log on 34
overriding application folder's working folder 289, 322
overriding view's working folder 290, 324
overview
change requests 13, 155
folders 41
labels 215
manual 4
projects 5
references 235
requirements 149
tasks 3
Toolbar 29
topics 16
version control 13
views 19
owner
defined 332
Owner requirement field 88

P

p
bco command option 323
command-line option 290
palette bar
charts 266
panes
defined 332
refreshing 9
Parent Branch Revision change request field 81
Parent Branch Revision file field 72
Parent ID change request field 81
Parent ID file field 73
Parent Requirement ID requirement field 88
Parent Revision change request field 81
Parent Revision file field 73
Parent Task ID task field 95
Parent Topic ID topic field 101
passwords
at command line 290
bco command 324
changing 283
Path file field 73
path names
platform differences 289, 290, 322, 324
paths
default location for reports 274
StarTeam.log 274
pattern bar
charts 266
patterns
changing on charts 265
Pending
task status 175
Percent Complete task field 96
Percentage Complete
tasks 176
performance
filters 53
permitted
process items 212
personal information
changing 283
personal options
at command line 315
defined 332
files 147
read and unread 208
read/unread 280
Visual Merge 114, 276
pie charts
exploding 267
pinning
file revisions to process items 213
Platform change request field 81
platform compatibility 289, 290, 322, 324
pop-up menus 8, 129, 158
defined 329
Toolbar system tray icon 32
Toolbar window 31
predecessors 175
preview
bco command 320
primary sort
defined 334
printing
audit entries 206
change requests 206
charts 264
compared files 123
detail reports 250
file properties 206
items 206
reports 250
tasks 206
topics 206
Priority
change requests 160
column in change request list 159
column in requirement list 150
tasks 175
Priority change request field 82
Priority requirement field 88
Priority task field 96
Priority topic field 101
process
items 211
rules 211
process items
active 212
change requests 211
linked at command-line 292, 300, 301
linking to file revisions 213
links 213
permitted 212
restrictions 212
setting 212
status changed at command-line 292, 300
tasks 211
product support 1
profiles
defined 332
folder settings 26
folder visibility 25
MPX 19, 20
views 25
program shortcuts
copying for Toolbar 36

creating for Toolbar 35
deleting 37
modifying properties 36
renaming 36
progress reports
command-line 290
Project audit log field 105
Project ID file field 73
projects
adding files 127
adding folders 44
adding tasks 174
defined 332
descriptions at command-line 295
keyword extensions at command-line 296
moving change requests 193
moving files 193
moving folders 193
moving tasks 193
moving topics 193
names at command-line 296
opening 20
overview 5
sharing change requests 195
sharing files 195
sharing folders 195
sharing tasks 195
sharing topics 195
shortcuts 22
understanding 5
promotion state configurations 23
promotion states
at command-line 297, 302, 305, 306, 308, 311, 313, 317
bco command 320
configuration defined 332
configuring folders or items to 201
defined 332
promotional model 332
properties 231, 332
comparing 203
connection 27
files 145
folders 46
labels 221
linked items 231
links 231
program shortcuts 36
tasks 179
topics 185
view 26
protocols
defined 332
purging
file status information 278
PVCS Revision file field 73
pwdfile
bco command option 324
command-line option 290

Q

q
command-line option 290
queries
access rights 65
applied to upper pane 55
copying 60
creating 56
defined 332
deleting 60
editing 60
selecting 56
status bar information 12
using 56

R

r
command-line option 301, 309
RC2_CBC 287, 320
RC2_CF 287, 320
RC2_ECB 287, 320
RC4 287, 320
read 208, 280
change requests 208
requirements 151
tasks 173, 208
topics 208
read items 208
Read Only change request field 82
Read Only file field 73
Read Only requirement field 89
Read Only task field 96
Read Only topic field 101
Read Status change request field 82
Read Status requirement field 89
Read Status task field 96
Read Status topic field 101
Read Status User List change request field 82
Read Status User List requirement field 89
Read Status User List task field 96
Read Status User List topic field 101
read-only
files 276
reference views defined 333
read-only files
at command-line 292, 301, 304, 310
bco command 324
read-only views
at command-line 298
read-write files
at command-line 292, 301, 304, 310
Reason for Check-In
history list 204
Recipient Count topic field 101
Recipient IDs topic field 101
Recipient Names topic field 102
reconfiguring
change requests 200
defined 333
files 200
folders 200
tasks 200
topics 200
views 22
recording
change requests 161
red
deleted text color 115, 122
reference counts
defined 333

Reference tab
change requests 161
description 11
files 129
requirements 152
tasks 172
topics 182
reference views
defined 333
references
created by adding items 241
created by branching views 239
created by manual sharing 244
floating downwards 242
floating upwards 242
floating upwards and downwards 243, 244
initial 238
items 236
overview 235
Reference tab 11
topics 185
refresh
defined 327, 333
Refresh Tools command (of Toolbar) 32
refreshing
audit entries 188
change requests 158
client 274
data in main window 9
external archives (PVCS and SourceSafe) 27, 129
file status 134
files 129
requirements 150
tasks 172
Toolbar 37
views 27, 129, 150, 158, 188
relational operators 65
for Boolean fields 66
for date/time fields 66
for enumerated type fields 66
for numeric fields 66
for text fields 65
relative operators 56
removals
confirming 273
remove
command at DOS prompt 314
removing
chart patterns 266
files at command-line 314
flags 208
folders 209
from version control (defined) 333
task attachments 179
renaming program shortcuts 36
reopening
change requests 165
replying
topics 185
reports
available 253
change request 168
command-line 290
database names 257
display field name 257
exporting data for use in other applications 251
output paths 274
printing 250
tables 257
templates 257
repository
defined 333
file statuses 24, 277
repository customization
defined 333
req
command-line option 292, 301
Requirement component
defined 333
requirement fields 84
requirement list
benefits 151
Created By 150
Created Time 150
Description 150
Locked By 150
Name 150
Number 150
Priority 150
sorting 150
Status 150
requirement trees
benefits 151
collapsing 151
expanding 151
requirements
at command line 292, 301
CaliberRM 154
creating 152
details 152
double-clicking 149
drag-and-drop 150
filters 151
Flagged Items 151
Grouped by Creator 151
Grouped by Status 151
history 152
I Am Responsible 151
linking 152
overview 149
read 151
refreshing 150
Show All 151
sorting 151
unread 151
using tree format 151
reset scope to local on folder change 273
resetting
filters 64
Resolved On change request field 82
resolving
change requests 163, 167
Resource Count task field 96
Resource IDs task field 96
Resource Names task field 96
resources
assigning 176
responding
topics 185
responses
contents 188
creating 185

defined 333
double-clicking 182
managing 181
sorting 183
See also topics.
Responsibility
change requests 162
Responsibility change request field 82
Responsibility task field 96
Responsible Count requirement field 89
Responsible IDs requirement field 89
Responsible Names requirement field 89
restoring
folder selection 273
shortcuts 274
Reviewed By requirement field 89
reviewing
change requests 163
label properties 221
process item links 213
Revised Description requirement field 89
revision comments
editing for files 204
Revision file field 73
Revision Flags change request field 82
Revision Flags file field 73
Revision Flags requirement field 89
Revision Flags task field 97
Revision Flags topic field 102
revision labels
cloning 219
command line 309
command-line 299, 304
copying 220
creating 219
defined 333
revision numbers
defined 333
Revision on Disk file field 74
revisions
as of 201
as of date (at command line) 299, 304, 307
attaching labels 224
checking in 127, 134
checking out 127, 136
checking out most recent 136
checking out past revisions 138
comments 179
comparing 111, 140
comparing change requests 203
comparing two 140
comparing two tasks 203
comparing two topics 203
comparing with working file 140, 141
configurations 201
creating revision labels for folders 222
creating revision labels for items 224
date/time 201
defined 333
detaching labels 225
history list 203
History tab 10
labeled 201
labels at command line 299
linked 229
merging 111
moving labels for folders 222
moving labels from one item revision to another 224
number at command line 299, 304, 307
reviewing contents of file revisions 140
selecting by label 225
selecting by query 56
rf
command-line option 301
right-clicking 129, 158
ro
bco command option 324
command-line option 292, 298, 301, 304, 310
rolling back
change requests 200
files 200
folders 200
tasks 200
topics 200
views 22
root folders
defined 333
Root Object ID change request field 82
Root Object ID file field 74
rotating
charts 264, 267
rp
bco command option 322, 324
command-line option 289, 290
rules
process 211
rw
command-line option 292, 301, 304, 310

S

s
command-line option 296, 315
saving
charts 264
shortcuts 22
task attachments 179
searching
audit entry fields 205
change request fields 205
file fields 205
task fields 205
text 116, 122
text in compared files 116, 122
topic fields 205
Visual Merge text 116
secondary sort
defined 333
selecting
labels 225
queries 56
Selection Version
link list column 229
sending
change requests via e-mail 206
item as shortcut 207
tasks via e-mail 206
topics via e-mail 206
server
disconnecting from 27
server configuration
defined 334

server description
for current view window 27
server-mode
command at DOS prompt 314
servers
at command line 315
command-line identification 296
defined 334
locking at command line 314
transferring data 21
unlocking at command line 314
Severity
change requests 161
Severity change request field 83
sf
command-line option 311
Share State change request field 83
Share State file field 74
Share State requirement field 90
Share State task field 97
Share State topic field 102
sharing
change requests 195
confirming operations 273
files 195
folders 195
tasks 172, 195
topics 195
Short Comment change request field 83
Short Comment file field 74
Short Comment requirement field 90
Short Comment task field 97
Short Comment topic field 102
shortcuts
defined 334
enabling compression 21
enabling encryption 21
for Toolbar 35
item 194
project 22
restoring 274
saving 22
sending via e-mail 207
Show All
audit log 188
change requests 160
requirements 151
topics 183
Show Unread Changes
change requests 160
showing
chart legends 264
simple charts 260, 262
Size file field 74
SMTP
defined 334
sorting
audit log 54, 187
change request list 54, 159
data in columns 9
defined 334
file list 54, 129
link list 54
requirement list 150
requirements 151
task list 54
tasks 173
topic list 54
topics and responses 183
splitter bar 121
square brackets convention 2
stand-alone
tasks 171
StarDisk
defined 334
StarTeamMPX 20
setting options 282
status bar information 282
stopping 283
StarTeamMPX Server 19
statistics
bco command 324
Status
column in requirement list 150
status bar
help for menus 12
queries 12
upper pane statistics 12
view configuration 12
Visual Merge 118
Status change request field 83
Status file field 74
Status requirement field 90
Status task field 97
Status topic field 102
statuses
change requests 157, 160
defined 334
determined by checksums 277
effects on checking files in and out 139
file updates 134
files 24, 277
files at command line 299, 300, 302, 305, 306, 308, 310, 311, 313, 314, 317
files in bco command 322
tasks 175
stcmd 285
add 292
add-folder 293
add-project 295
add-view 297
apply-label 298
ci 299
co 301
delete-local 304
diff 305
dsc 307
hist 308
label 309
lck 310
list 311
local-mkdir 312
move-status 312
remove 314
server-mode 314
set-personal-options 315
update-status 316
stop
command-line option 291
stopping
StarTeamMPX 283
Storage Type file field 74

storing
versions 128
structures
folders 43
subdirectories
See folders and child folders.
subfolders
See folders and child folders.
submitting
change requests 161, 166
subtasks
See tasks.
suggestion
change request type 161
support 1
suppressing
line numbers at command line 307
swiveling
charts 267
Sync Branch Version file field 74
Sync Content Version file field 74
Sync Known file field 75
Sync Local Size file field 75
Sync Local Time Stamp file field 75
Sync MD5 file field 75
Sync On Path To Root file field 75
syncdb
file statuses 277
Synopsis
change requests 162
Synopsis change request field 83
syntax
command-line 286
syntax conventions
mutually-exclusive choices 2
options 2
variables 2
system tray
notification 281
system tray icon (for Toolbar) 32
hiding 32
system tray notification 14, 281

T

t
bco command option 324
command-line option 307
tab settings
Visual Merge 118
tab stops
option at command line 307
table
reports 257
tabs
Audit Trail 187
Target 1 Class ID audit log field 105
Target 1 Object ID audit log field 105
Target 1 Revision Time audit log field 105
Target 2 Class ID audit log field 105
Target 2 Object ID audit log field 105
Target 2 Revision Time audit log field 106
Target 3 Class ID audit log field 106
Target 3 Object ID audit log field 106
Target 3 Revision Time audit log field 106
task
command-line option 292, 301
pane 171
task component
defined 334
Task Duration task field 97
task fields 90
task lists
benefits 173
controlling 171
sorting 54
Task Name task field 97
Task Number task field 97
Task Origin task field 97
task trees 171
benefits 172
collapsing 172
controlling 171
expanding 172
icons 172
Task Type task field 97
taskbar button (for Toolbar window) 32
tasks
access rights for labels 226
adding notes 178
adding task to project 174
affected by change in view 193
assigning resources 176
at command line 292, 301
attaching labels 224
attachments 178
charts 259
comparing properties 203
comparing revisions 203
configuring 200
copying revision labels 220
creating 174
creating revision labels 224
custom fields 178
deleting 209
detaching labels 225
details 172
do not level 176
double-clicking 172
drag-and-drop 172
filters 173
finding 205
finished 211
flagging 208
history 172
label overview 215
linking 172, 227
managing 171
marking read or unread 208
menu 172
merging 205
milestone 175
moving 193
moving labels 224
needs attention 176
notifications 14, 281
operations on 191
overview 3
Percentage Complete 176
predecessor 175
printing 206
priorities 175

process items 211
properties 179
read 173
refreshing 172
removing attachments 179
revision comments 179
rolling back 200
saving attachments 179
searching fields for specific data 205
selecting by label 225
selecting by query 56
sending e-mail 206
sharing 172, 195
shortcut 194
sorting 173
statuses 175
time estimates 176
unread 173
using tree format 172
without Microsoft Project integration 171
work records 177
workstation features 280
TCP/IP
defined 334
team members
assigned to tasks 176
defined 334
technical support 1
templates
changing reports 257
Test Command
column in change request list 159
defined 334
Test Command change request field 83
testing
change requests 165
text
relational operators 65
searching compared files 122
topics and responses 188
text files
comparing 111
defined 334
differences 305
merging 111
reports 251
searching compared files 116
searching in Visual Merge 116
The 3
third-party merge application 303
threaded conversation
defined 334
three-dimensional charts 264, 267
three-way merge 111
tilting
charts 267
time stamps
bco command 324
checking out 275
command line 304
defined 334
history list 203
status determined by 134
times
estimated for tasks 176
local 273
UTC 273
time-series charts 263
timing statistics
bco command 324
tip revisions
defined 335
Title topic field 102
titles
editing charts 264, 267
Toolbar
About command 32
Add Tool command 32
adding tools 35
bco command 319
caching user name and password 34
closing window 38
collapsing window 38
commands on system tray icon's pop-up menu 32
commands on window's pop-up menu 31
copying existing shortcuts 36
creating new shortcuts 35
customizing 35
Delete Tool command 32
deleting tools 37
displaying About dialog 39
displaying help 38
displaying window 31
displaying window at startup 31
ending process (from Task Manager) 39
Exit command 32
exiting 39
expanding window 38
features 29
Help command 32
hiding system tray icon 32
hiding taskbar button 32
launching tools 35
logging on as default user 33
logging on as different user 34
logging onto servers 33
managing window 37
minimizing window 38
Modify Tool command 32
modifying properties of tools 36
Open command 32
Open Tools Folder command 32
opening window 30
overview 29
Refresh Tools command 32
refreshing 37
renaming tools 36
resizing window 38
starting 30
system tray icon 32
user interface 31
using same user name and password for each server 34
varying functionality 31
viewing cached server/user associations 33
window 31
toolbar buttons
Visual Diff 119
toolbars
chart window 264
Visual Merge 118
tools

adding to Toolbar 35
charts 264
topic fields 98
topic list
sorting 54
topic lists
benefits 183
Topic Number topic field 102
topic trees
benefits 182
collapsing 182
defined 335
expanding 182
topics
access rights for labels 226
affected by change in view 193
attaching labels 224
attachments 184
Author 183
charts 259
comparing properties 203
comparing revisions 203
configuring 200
contents 188
copying revision labels 220
creating 184
creating revision labels 224
custom fields 184
Date Range 183
defined 335
deleting 209
detaching labels 225
details 182
double-clicking 182
filters 183
finding 205
flagging 208
history 182
label overview 215
linking 182, 227
managing 181
marking read or unread 208
merging 205
moving 193
moving labels 224
notifications 14, 281
operations on 191
overview 16
printing 206
properties 185
rolling back 200
searching fields for specific data 205
selecting by label 225
selecting by query 56
sending e-mail 206
sharing 195
shortcuts 194
Show All 183
sorting 183
using tree format 182
view references 185
workstation features 280
transferring
data 21
Tree Display button 151, 182
tree format 172
requirements 151
tasks 172
topics 182
trees
task icons 172
ts
bco command option 324
command-line option 304
two-dimensional charts 264, 267
two-way difference 111
Type
change requests 160
column in audit log 187
Type change request field 83
Type requirement field 90
Type topic field 102
types
charts 264
views at command-line 297

U

u
command-line option 293, 301, 304, 310
unfreezing
labels 220
UNIX
executable bit 132
Unknown
defined 335
file status 139
unlocking
defined 335
files 135
files at command-line 293, 301, 304, 310
files on check-in 276
items 202, 282
servers at command line 314
unread 208, 280
change requests 160, 208
requirements 151
tasks 173, 208
topics 208
unread items 208
update-status
command at DOS prompt 316
updating
file statuses 134
updating statuses
files at command-line 316
upper pane
controlling 8
creating filters for 62
explained 7
filters 61
queries 55
sorting 9
useCA
bco command option 324
User
column in audit log 187
User audit log field 106
User's Guide
overview 4
users
assigned to tasks 176

defined 335
using
filters 62
keywords 144
queries 56
task pane 171
Toolbar 29
version control 13
UTC times 273
utility
bulk check-out 319

V

variant views
See branching views. 335
Vault Branch Version file field 75
vaults
defined 335
vb
bco command option 324
vd
command-line option 299, 304, 307, 309
verbose
bco command 324
Verified On change request field 83
verifying
change requests 164, 168
Version change request field 83
version control
defined 335
overview 13
version control systems
defined 335
Version file field 75
version labels
See revision labels.
version numbers
See revision numbers.
Version requirement field 90
Version task field 98
Version topic field 102
versions
comparing 111
grouping by label 128
managing 127
merging 127
order in Visual Diff 121
reports available 253
storing non-permanent 128
storing permanent 128
vertical
chart grids 264
vertical bars convention 2
View
history list 203
link list column 229
View audit log field 106
View change request field 84
View file field 75
view labels 216
at command line 297, 302, 305, 317
bco command 320
build labels 216, 217
command line 309
command-line 299, 304
copying 217
creating 217
defined 335
view window
defined 335
explained 7
view windows
current connection properties for 27
viewing
cached server/user associations 33
charts 264, 267
file properties 145
item references 236
link properties 231
linked items' properties 231
task properties 179
topic references 185
viewmerge
command and options 114
views
access rights for labels 226
adding files 127
adding folders 44
alternate working folder properties 24
alternate working folders 44
bco command option 323
branching at command line 297
command-line option 290
configuration as of date at command line 297, 302, 305, 306, 308, 311, 312, 313, 317
configuration as of date in bco command 320
configuration name at command-line option 297, 302, 305, 306, 308, 311, 312, 313, 317
configuration name in bco command 320
configuration shown in status bar 12
configuration state at command-line option 297, 302, 305, 306, 308, 311, 313, 317
configuration state in bco command 320
date/time configurations 23
defined 335
deleting labels 221
derived at command-line 298
descriptions at command-line 298
files 130
labeled configurations 23
labels at command line 299
moving change requests 193
moving files 193
moving folders 193
moving tasks 193
moving topics 193
names at command-line 298
non-excluded files 130
overriding working folder at command-line 289, 290
overriding working folder in bco command 322, 324
overview 19
profiles 25
promotion state configurations 23
properties 26
read-only at command-line 298
reconfiguring 22
references created by branching 239
refreshing 27, 129, 150, 158, 188
rolling back 22
sharing change requests 195
sharing files 195

sharing folders 195
sharing tasks 195
sharing topics 195
storing file statuses 24
types at command-line 297
Visual Diff 140, 141
changing color codes 122
changing file encodings 120
compared to Visual Merge 111
comparing files or versions 111
default color codes 121
default comparison utility 280
defined 335
differences 122
finding differences 116, 122
finding matches 116, 122
finding text in compared files 116, 122
main window 119
matches 122
menu commands 119
merging files 123
order of files 121
splitting window 121
toolbar buttons 119
two-way difference 111
Visual Merge 142
am option 114
amt option 114
at option 114
changing file encodings 115
colors 118
compared to Visual Diff 111
comparing and merging revisions 111
conflicts 117
customizing 118
default merge utility 279
defined 335
differences 117
example 111
exiting 117
font 118
iw option 114
line numbers 118
main window 115
main window appearance 118
personal options 114
searching text 116
StarTeam personal options 276
status bar 118
tab settings 118
three-way merge 111
toolbar 118
used as a stand-alone utility 113
windows 118
vl
command-line option 293, 299, 301, 304, 307, 310
vn
command-line option 299, 304, 307
volume statistics
bco command 324
vp
command-line option 310

W

w
command-line option 307
warnings
confirming 273
whitespace
ignoring 124
option at command line 306, 307
window (for Toolbar) 31
windows
chart 264
refreshing 9
Visual Diff 119, 121
Visual Merge 115, 118
wizards
New Folder 44
Work Around
column in change request list 159
Work Around change request field 84
work arounds
defined 336
Work Record Count task field 98
work records 177
workflow
affected by change in view 193
Working File Exists file field 76
Working File Size file field 76
working files
comparing 141
comparing to latest revision 141
comparing to revision 140
defined 336
deleting at command line 314
detecting changes to 277
working folders 42
alternate 44, 48
alternate paths 24
changing 48
creating 43, 44
creating at command line 312
defaults 43
overriding at command line 289, 290
overriding in bco command 322, 324
Working Time Stamp file field 76
workstations
change request features 280
file features 275
task features 280
topic features 280
transferring data 21

X

x
command-line option 291

Z

z-cluster charts 264
zooming
charts 264